EXHIBIT M

IN THE UNITED STATES DISTRICT COURT
EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

SOFTWARE RIGHTS ARCHIVE, LLC v. GOOGLE INC., YAHOO! INC., IAC SEARCH & MEDIA, INC., AOL LLC, AND LYCOS, INC.	Civil Case No. 2:07-cv-511 (CE)

APPENDIX TO DEFENDANTS' P. R. 3-3 DISCLOSURE

I. INTRODUCTION

This Appendix includes Tables and disclosures to amend and supplement Defendants' Patent Rule 3-3 Disclosures (hereinafter "Invalidity Contentions") as specifically set forth below. Except as specifically stated, nothing in this Appendix is intended to waive or otherwise limit the positions and arguments set forth in Defendants' Invalidity Contentions.

Defendants' Invalidity Contentions and this Appendix are based in whole or in part on Defendants' present understanding of the asserted claims and SRA's apparent positions as to the scope of the asserted claims as applied in its P. R. 3-1 disclosures. Accordingly, Defendants' Invalidity Contentions and this Appendix (including any attached invalidity claim charts) reflect, to the extent possible, SRA's expected alternative and potentially inconsistent positions as to claim construction and scope. In addition, SRA has yet to disclose the details of its basis for its infringement contentions and its basis for contending that the '494 and '571 patents have written description support in application U.S. Patent Application No. 08/076,658. Accordingly, Defendants reserve the right to further amend or supplement their Invalidity Contentions and this Appendix.

II. IDENTIFICATION OF PRIOR ART PURSUANT TO P. R. 3-3(A)

Pursuant to P. R. 3-3(a), and subject to Defendants' reservation of rights, Defendants identify each supplemental item of prior art that anticipates or renders obvious one or more of the asserted claims in Table App-2B below. Table App-2B further supplements Table 2.

Table App-2A: Items Used and/or Offered for Sale

Defendants identify that electronic information previously produced in this action with production numbers DEF0016248-DEF0016354 relates to system(s) and method(s) that were used and/or offered for sale before the invention of the '352 patent. Defendants will produce source code for UCINET once a suitable protective order for source code is in place. Defendants

reserve the right to supplement their contentions regarding the system(s) and method(s) disclosed in such electronic information and as identified based on further discovery.

Table App-2B: Prior Art Publications and Items Used and/or Offered for Sale

Primary Author or Publisher	Reference Title	Publication/ Use Date	Herein Referenced As
Brodda, B. & Karlgren, H.	"Citation Index And Measures Of Association In Mechanized Document Retrieval," Kval Pm 295 (1967). Report No. 2 To The Royal Treasury. Published By Sprakforlaget Skriptor.	1967	Brodda & Kalgren 1967 (or Brodda 1967)
Schatz, B. & Hardin, J.	"NCSA Mosaic and the World Wide Web: Global Hypermedia Protocols for the Internet," Science 265:895-901 (1994)	1994	Schatz 1994
Cleveland, D.	"An n-Dimensional Retrieval Model," J. Am. Soc. Inf. Sci., pp. 342-47 (1976)	1976	Cleveland 1976
Crouch, D. et al.	"The Use Of Cluster Hierarchies In Hypertext Information Retrieval," Hypertext '89 Proceedings, SIGCHI Bulletin, pp. 225-237 (Nov. 1989)	1989	Crouch 1989
Salton, G. & Buckley, C.	"Approaches to Text Retrieval for Structured Documents," TR 90-1083, Department of Computer Science, Cornell University (1990)	1990	Salton, 1990 (or Salton & Buckley 1990)
Salton, G. & Buckley, C.	"On the Automatic Generation of Content Links in Hypertext," TR89-1993. (Department of Computer Science, Cornell University), (1989)	1989	Salton, 1989 (or Salton & Buckley 1989)
Korfhage, R.	"Query Enhancement by User Profiles" (1983)	1983	Korfhage, 1983
Baase, S.	Computer Algorithms: Introduction to Design and Analysis, 2nd Edition, Addison-Wesley Publishing Co. (1988)	1988	Baase 1988
Can, F.	"A Dynamic Cluster Maintenance System for Information Retrieval," ACM, Vol. 6, p. 123 (1987)	1987	Can, 1987
Botafogo, R.	"Cluster Analysis for Hypertext Systems," ACM SIGIR '93, Vol. 6,	1993	Botafogo 1993

Primary Author or Publisher	Reference Title	Publication/ Use Date	Herein Referenced As
	116-125 (1993)		
Botafogo, R.	"Identifying Aggregates in Hypertext Structures"	1991	Botafogo 1991
Joachims, T et al.,	"WebWatcher: Machine Learning and Hypertext" . Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments	1995	Joachims 1995
Caplinger, M.	"Graphical Database Browsing" ACM, p. 113-121	1986	Caplinger 1986

III. INVALIDITY CONTENTIONS CONCERNING U.S. PATENT NO. 5,544,352

A. Disclosure of Invalidity Due to Anticipation Pursuant to P. R. 3-3(b) and (c)

Table 3 is supplemented by the addition of Table App-3 which includes the following patents and publications which are prior art under at least 35 U.S.C. §§ 102(a), (b), (e), and/or (g).

Table App-3: Patents and Printed Publications Anticipating the Asserted Claims of the '352 Patent

Exhibit A Chart	Prior Art
Ex A-58	Brodda & Karlgren 1967
Ex A-59	Cleveland 1976
Ex A-60	Baase 1988
Ex A-61	Crouch 1989

B. Disclosure of Invalidity Due to Obviousness Pursuant to P. R. 3-3(b) and (c)

The asserted claims of the '352 Patent are invalid as obvious under 35 U.S.C. § 103.

1. Obviousness Combinations

Defendants withdraw the combination of references previously presented in Exhibit C-1 of their Invalidity Contentions and add Table App-5. In response to SRA's request for clarification, Table App-5 provides specific combinations of references that render obvious the asserted claims of the '352 Patent:

Table App-5: References Rendering Obvious Asserted Claims of the ’352 Patent

Combination	Claims of the ’352 Patent Rendered Obvious by the Combination
Salton (1963) Salton & McGill (1983)	26-32, 36-40, 45
+ Gelbart (1991) or Fox (Smart 1983) + Cleveland (1976), Korfhage (1983), or Burt (1991)	26-42, 44, 45
+ Gelbart (1991) or Fox Smart (1983)	26-32, 35-40, 45
+ Brodda (1967)	26-32, 36-40, 45
+ Brodda (1967) + Cleveland (1976), Burt (1991), or Korfage (1983)	26-34, 36-42, 44, 45
+ Garfield (1979)	26-32, 36-40, 45
Salton (1963) Salton (1971)	26, 29-32, 36-40, 45
+ Cleveland (1976), Burt (1991), or Korfage (1983)	26, 29-33, 36-42, 44, 45
Salton (1963) Fox Thesis (1983)	26-42, 44, 45
+ Cleveland (1976), Burt (1991), or Korfage (1983)	26-42, 44, 45
Salton (1963) Salton & Buckley (1990)[1] or Salton & Buckley (1989)	26-32, 34-40, 45
Salton (1963) Pinski (1976)	26-27, 29-32, 39, 41, 42, 44, 45
Garner (1967) Salton & McGill (1983)	26-32, 36-40, 45
+ Can (1987)	26-32, 36-40, 45
+ Cleveland (1976), Burt (1991), or Korfage (1983)	26-32, 36-42, 44
+ Cleveland (1976), Burt (1991), or Korfage (1983) + Gelbart (1991) or Fox Smart (1983)	26-42, 44, 45
Garner (1967) Thompson (1989)	26-32, 36, 37, 41, 45
Garner (1967)	26, 29-32, 41, 45

[1] Salton & Buckley (1990) discloses at least the limitations of claims 34, 35 (see passim, including p. 2-4, Fig. 2), 38 (p. 5), 39 (p. 4-5), and 42 (p. 3-4). These disclosures are applicable to all combinations that include Salton & Buckley (1990) listed in Table App-5.

Combination	Claims of the ’352 Patent Rendered Obvious by the Combination
Frisse (1988)	
Salton (1971) Salton & McGill (1983)	26-32, 36-40, 45
Fox Thesis (1983) Salton & McGill (1983)	26-42, 44, 45
+ Cleveland (1976)	26-42, 44, 45
+ Gelbart (1991) or Fox Smart (1983)	26-42, 44, 45
+ Cleveland (1976), Burt (1991), or Korfhage (1983)	26-42, 44, 45
+ Garner (1967), Garfield 1979	26-42, 44, 45
+ Fox Smart (1983) + Pinski (1976) + Cleveland (1976), Burt (1991), or Korfhage (1983)	26-42, 44, 45
Fox Thesis (1983) Shepherd (1990)	26-42, 44, 45
Fox Thesis (1983) Fox Collections (1983)	26-42, 44, 45
+ Gelbart (1991) or Fox Smart (1983)	26-42, 44, 45
Fox Thesis (1983) Fox Smart (1983)	26-42, 44, 45
+ Cleveland (1976), Burt (1991), or Korfhage (1983)	26-42, 44, 45
+ Garfield 1979	26-42, 44, 45
+ Can (1987)	26-42, 44, 45
+ Salton (1990)	26-42, 44, 45
Fox Thesis (1983) Kochtanek (1982)	26-42, 44, 45
Fox Thesis (1983) Thompson (1989)	26-42, 44, 45
Fox Thesis (1983) Garner (1967)	26-42, 44, 45
Fox Thesis (1983) Burt (1991)	26-42, 44, 45
+ Gelbart (1991) or Fox Smart (1983) + Cleveland (1976), Burt (1991), or Korfhage (1983)	26-42, 44, 45
Fox Thesis (1983) Berk (1991)	26-42, 44, 45

Combination	Claims of the ’352 Patent Rendered Obvious by the Combination
Fox Thesis (1983) Tapper (1982)	26-42, 44, 45
+ Cleveland (1976), Burt (1991), or Korfhage (1983)	26-42, 44, 45
Fox Thesis (1983) Fox (1985)	26-42, 44, 45
Fox Thesis (1983) Gelbart (1991)	26-42, 44, 45
Fox Thesis (1983) Cleveland (1976)	26-42, 44, 45
Fox Thesis (1983) Rose (1991)	26-42, 44, 45
Fox Thesis (1983) Korfage (1983)	26-42, 44, 45
Fox Thesis (1983) Salton & Buckley (1990) or Salton & Buckley (1991)	26-42, 44, 45
Fox Thesis, Garfield 1979, Pinski 1976, Conklin 1987, Berners-Lee 1989	26-42, 44, 45
Salton & McGill (1983) Thompson (1989)	26-32, 36-40, 45
Salton & McGill (1983) Kochtanek (1982)	26-32, 36-40, 45
Salton & McGill (1983) Shepherd (1990)	26-32, 36-40, 45
Salton & McGill (1983) Brodda (1967)	26-32, 36-40, 45
+ Gelbart (1991) or Fox Smart (1983) + Cleveland (1976), Burt (1991), or Korfage (1983)	26-42, 44, 45
Salton & McGill (1983) Burt (1991)	26-33, 36-42, 44, 45
+ Gelbart (1991) or Fox Smart (1983) + Cleveland (1976), Burt (1991), or Korfage (1983)	26-42, 44, 45
Salton & McGill (1983) Tapper (1982)	26-32, 36-40, 45
+ Cleveland (1976), Burt (1991), or Korfage (1983)	26-33, 36-42, 44, 45
Salton & McGill (1983) Fox (1985)	26-32, 34-40, 45
Salton & McGill (1983) Salton & Buckley (1990) or Salton & Buckley (1989)	26-32, 34-40, 45

Combination	Claims of the ’352 Patent Rendered Obvious by the Combination
Fox Smart (1983) Salton & McGill (1983)	26-32, 34-41, 45
Salton & McGill (1983) Gelbart (1991)	26-32, 34-40, 45
Salton & McGill (1983) Rose (1991)	26-32, 36-41, 45
Frisse (1988) Shepherd (1990)	26-32
+ Rose (1991)	26-34, 36-42, 45
Frisse (1998) Nielsen (1990(b))	26-32, 35-40, 45
Belew (1986) Rose (1991)	26-34, 36-39, 41, 42, 44, 45
Kaplan 891 Patent + Lucarella 1990 + Conklin (1988)	26-32, 34, 36-37, 45
Lucarella 1990 + Turtle 1991 + Croft & Turtle 1991	26-32, 34, 36-37, 45
Rose 1989, Rose 1991, Tapper 1982, Conklin (1988)	26-34, 36-39, 41, 42, 44, 45

In addition, each claim is obvious in view of cited references in combination with the general knowledge in the art. Additional specific combinations are described in Ex. C-2, C-3, C-4, C5, and C-6 of Table 5 in the Invalidity Contentions of January 23, 2009.

With respect to the references and combinations disclosed herein, Defendants incorporate by reference Section III.B.1 (except for Exhibit C-1) of their Invalidity Contentions of January 23, 2009. Defendants further reference the following:

Exhibit A Chart	Prior Art
Ex A-62	Can 1987
Ex A-63	Salton & Buckley (1991)
Ex A-64	Salton & Buckley (1990)
Ex A-65	Korfhage 1983

2. Motivation to Combine

With respect to the references and combinations disclosed herein, Defendants incorporate by reference Section III.B.2 of their Invalidity Contentions of January 23, 2009.

IV. INVALIDITY CONTENTIONS CONCERNING U.S. PATENT NO. 5,832,494

B. Disclosure of Invalidity Due to Anticipation Pursuant to P. R. 3-3(b) and (c)

Table 6 is supplemented by the addition of Table App-6 which includes the following patents and publications are prior art under at least 35 U.S.C. §§ 102(a), (b), (e), and/or (g).

Table App-6: Patents and Printed Publications Anticipating the Asserted Claims of the '494 Patent

Exhibit App-D Chart	Prior Art
Ex D-58	Brodda & Karlgren 1967
Ex D-59	Baase 1988
Ex D-60	Crouch 1989
Ex D-61	Botafogo 1993
Ex D-64	Botafogo 1991
Ex D-65	Joachims 1995

C. Disclosure of Invalidity Due to Obviousness Pursuant to P. R. 3-3(b) and (c)

The asserted claims of the '494 Patent are invalid as obvious under 35 U.S.C. § 103.

1. Obviousness Combinations

Defendants withdraw the combination of references previously presented in Exhibit F-1 of their Invalidity Contentions and add Table App-8. In response to SRA's request for clarification, Table App-8 provides specific combinations of references that render obvious the asserted claims of the '494 Patent:

Table App-8: References Rendering Obvious Asserted Claims of the '494 Patent

'494 Patent Combinations and Asserted Claims

Combination	Claims Rendered Obvious By The Combination
Nielsen (1990b) Lucarella (1990	1-3, 5, 7-16, 18-21, 23-25, 31-33

Combination	Claims Rendered Obvious By The Combination
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Schatz (1994), Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Shepherd (1990) or Salton & McGill (1983)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Nielsen (1990b) Rose (1991)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Schatz (1994) or Doyle US 5838906, +Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Nielsen (1990b) Belew (1986)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Schatz (1994) or Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Nielsen (1990b) Brodda (1967)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33

Combination	Claims Rendered Obvious By The Combination
+ Schatz (1994) or Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Rose (1991) Belew (1986)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 12-16, 18-21, 23-25, 31-33
+ Schatz (1994) or Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 33
Rose (1991) Brodda (1967)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 12-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or Schatz (1994)	1-3, 5, 12-16, 18-21, 23-25, 33
Frisse (1988) Lucarella (1990)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Schatz (1994) or Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Frisse (1988) Rose (1991)	1-3, 5, 7-16, 18-21, 23-25, 33

Combination	Claims Rendered Obvious By The Combination
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Schatz (1994) or Doyle US 5838906 Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Frisse (1988) Belew (1986)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Schatz (1994) or Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Frisse (1988) Brodda (1967)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Schatz (1994) or Doyle US 5838906 + Berners Lee 1989, Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
Rose (1991) + Berners-Lee (1989)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993) or Croft & Turtle (1989)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Schatz (1994) or Doyle US 5838906 + Kaplan 1995	1-3, 5, 7-16, 18-21, 23-25, 33
+ Guinan (1992) or Weiss (1996) or Salton (1971) or Baase (1988)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Pitkow (1994) or Alain (1992) or	1-3, 5, 7-16, 18-21, 23-25, 33

Combination	Claims Rendered Obvious By The Combination
Conklin (1987) or Conklin (1988) or Fox/Envision (1993)	
Salton & McGill (1983) Garner (1967)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Brodda (1967)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Salton (1963)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Salton (1963) Brodda (1967)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Salton (1971)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Fox Thesis (1983)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Thompson (1989),	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Pinski (1976) Garfield (1979)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton & McGill (1983) Pinski (1976) Fox SMART (1983) Fox Thesis (1983)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Botafogo (1992) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Baase (1988), Pinski (1976)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Burt (1991), UCINET, Pinski (1976), Kommers (1990), LA Times	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Weiss (1996)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Botafogo (1993), Botofogo (1991)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Botafogo (1993)+ Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Baase (1988), Pinski (1976)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Burt (1991), UCINET, Pinski (1976), Kommers (1990) , LA Times	1-3, 5, 7-16, 18-21, 23-25, 31-33

Combination	Claims Rendered Obvious By The Combination
+ Weiss (1996)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Botafogo (1991)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Betrabet (1993) Betrabet Thesis (1993) Berk (1991) Fox (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Chen Thesis (1992), Chen (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Botafogo (1992), Pinski (1976), Guinan (1990)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Botafogo (1993), Conklin (1988)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992) or Salton (1988) or Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Botafogo (1992), Alain (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Shepherd (1990), Guinan (1990)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Betrabet (1993) Betrabet Thesis (1993) Berk (1991) Baase (1988) Shepherd (1990) Kommers (1990) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
Betrabet (1993) Betrabet Thesis (1993) Berk (1991) Burt (1991) UCINET Kommers (1990) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
Garfield (1979) Pinski (1976) Fox/Envision (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 33

Combination	Claims Rendered Obvious By The Combination
+ Croft (1993)	1-3, 5, 7-16, 18-21, 23-25, 33
Thompson (1989) Turtle (1991) Croft & Turtle (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 33
+ Croft (1993), Croft & Turtle (1989)	1-3, 5, 7-16, 18-21, 23-25, 33
Dunlop (1991) Frei & Steiger (1992) Frei & Steiger (1995) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Kochtanek (1982) or Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Croft (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
Salton (1988), Rose (1991),	1-3, 5, 12-16, 18-21, 23-25, 33
+ Thompson (1989) or Guinan (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
+ Frei & Steiger (1992)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Fox Thesis (1983) Thompson (1989)	1-3, 5, 7-16, 18-21, 23-25, 33
+ Pinski (1976), Garfield (1979), Garner (1967)	1-3, 5, 7-16, 18-21, 23-25, 33
Fox Thesis (1983) Garner (1967)	1-3, 5, 12-16, 18-21, 23-25, 33
Fox Thesis (1983) Nielsen (1990b)	1-3, 5, 7-16, 18-21, 23-25, 31-33
Pirolli (1996), Weiss (1996) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
Crouch (1989) Shepherd (1990) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1-3, 5, 7-16, 18-21, 23-25, 31-33
Joachims (1995)	1-3, 5, 7-16, 18-21, 23-25, 31-33

Combination	Claims Rendered Obvious By The Combination
+ Pinkerton (1994), Doyle US 5,838,906, Conklin (1988), Mauldin US 5,748,954, + Schatz (1994)	

In addition, each claim is obvious in view of cited references in combination with the general knowledge in the art. Knowledge and use of the internet is exemplified by at least the following references: Conklin, 1987; Berners-Lee, 1989; Krol, 1994; Pinkerton, 1994; LA Times; Doyle U.S. 5,838,906, Maudlin, Mauldin US 5,748,954, Shatz 1994, and Nielson 1990b.

Additional specific combinations are described in Ex. F-2, F-3, F-4, F-5, F-6, and F-7 of Table 8 in the Invalidity Contentions of January 23, 2009.

With respect to the references and combinations disclosed herein, Defendants incorporate by reference Section IV.C.1 (except for Exhibit F-1) of their Invalidity Contentions of January 23, 2009. Defendants further reference the following:

Ex D-62	Schatz

2. Motivation to Combine

With respect to the references and combinations disclosed herein, Defendants incorporate by reference Section IV.C.2 of their Invalidity Contentions of January 23, 2009.-

V. INVALIDITY CONTENTIONS CONCERNING U.S. PATENT NO. 6,233,571

B. Disclosure of Invalidity Due to Anticipation Pursuant to P. R. 3-3(b) and (c)

Table 9 is supplemented by the addition of Table App-9 which includes the following patents and publications are prior art under at least 35 U.S.C. §§ 102(a), (b), (e), and/or (g).

Table App-9: Patents and Printed Publications Anticipating the Asserted Claims of the ’571 Patent

Ex G-79	Botafogo 1993
Ex G-80	Crouch 1989
Ex G-82	Botafogo 1991
Ex G-83	Joachims 1995

C. Disclosure of Invalidity Due to Obviousness Pursuant to P. R. 3-3(b) and (c)

The asserted claims of the ’571 Patent are invalid as obvious under 35 U.S.C. § 103.

1. Obviousness Combinations

Defendants withdraw the combination of references previously presented in Exhibit I-1 of their Invalidity Contentions and add Table App-11. In response to SRA’s request for clarification, Table App-11 provides specific combinations of references that render obvious the asserted claims of the ’571 Patent:

Table App-11 provides specific and exemplary combinations of references that render obvious the asserted claims of the’571 Patent:

Table App-11: References Rendering Obvious Asserted Claims of the ’571 Patent

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
Botofago (1992) Pitkow (1994)[2] Conklin (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22

[2] With respect to claims 1, 3, 4, and 11, Pitkow 1994 discloses a source map as previously shown for claim 22 in Ex. G-79.

Combination	Claims of the '571 Patent Rendered Obvious by the Combination
Botofago (1992) Baase (1988) Conklin (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Botofago (1992) Baase (1988) Burt 1991, UCINET + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Frei & Steiger 1992, Frei & Steiger	1, 3-22
+ Pitkow (1994), Conklin (1988)	1, 3-22
+ Pinski (1976), LA Times	1, 3-22
+ Crouch (1989)	1, 3-22
+ Caplinger (1986), Conklin (1988)	1, 3-22
Botofago (1993) Conklin (1988), Pitkow (1994), or Caplinger (1986) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Baase (1988)	1, 3-22
+ Frei & Steiger 1992, Frei & Steiger 1995	1, 3-22
+ Burt 1991, UCINET, Botofago (1992)	1, 3-22
+ Crouch (1989)	1, 3-22
+ Pinski (1976) , LA Times	1, 3-22
+ Botafogo (1991)	1, 3-22
Garfield (1979) Pinski (1978) Fox/Envision (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
Rose (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Tapper (1982), LA_Times	1, 3-22
+ Belew (1986)	1, 3-22
+ Thompson (1989), Croft & Turtle (1989)	1, 3-22
+ Caplinger (1986), Conklin (1988), Netcarta	1, 3-22
Conklin (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-4, 22
Envision Garfield (1979) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-6, 8-16, 19-22
Envision Fox Thesis (1983) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Nielsen (1990) Nielsen (1990(b)) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ McKee (1994), Pitkow (1994)	1, 3-22
+ Frisse 1998	1, 3-22
+ Frei & Steiger 1992, Frei & Steiger 1995	1, 3-22
+ Botafogo 1992, Conklin (1988)	1, 3-22
+ Botafogo 1993, Conklin (1988)	1, 3-22
+ NetCarta, Conklin (1988), Caplinger (1986),	1, 3-22
+ Belew 1986, Rose (1991)	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
+ Brodda (1967)	1, 3-22
Thompson (1989) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Turtle (1991), Croft & Turtle (1991), Croft & Turtle (1989)	1, 3-22
Frisse (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Lucarella 1990	1, 3-22
+ Rose (1991)	1, 3-22
+ Frei & Steiger 1992, Frei & Steiger 1995	1, 3-22
+ Thompson (1989), Croft & Turtle (1989), Kaplan ’891 Patent	1, 3-22
+ Conklin (1988), NetCarta, Caplinger (1986)	1, 3-22
+ Frisse/Cousins, Crouch (1989)	1, 3-22
Frei & Steiger 1992, Frei & Steiger 1995 + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Caplinger (1986), Conklin (1988), Netcarta	1, 3-22
Garner (1967) Salton (1963) Salton & McGill (1983) Fox/Envision (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Shepherd (1990) Garfield (1979) Fox Thesis (1983) Fox/Envision (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
US 5,838,906, or Mauldin US 5,748,954	
Kaplan 891 Patent + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Kaplan 891 Patent Thompson (1989) Turtle (1991) Croft & Turtle (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Kochtanek (1982) Garfield (1979) Shepherd (1990) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Kochtanek (1982) Fox Thesis (1983) Fox/Envision (1993) Fox (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-9, 12-22
Croft (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	5-9, 11-21
+ Salton (1988), Turtle (1991), Croft & Turtle (1991)	5-9, 11-21
Lucarella (1990) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Lucarella (1990) Kaplan 891 Patent + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Lucarella (1990)	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
Turtle (1991) Croft & Turtle (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	
Betrabet (1993) Betrabet Thesis (1993) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Betrabet (1993) Betrabet Thesis (1993) Berk (1991) Fox (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Chen Thesis (1992), Chen (1992)	1, 3-22
+ Botafogo (1992), Pinski (1976), ,Guinan (1990)	1, 3-22
+ Botafogo (1993), Conklin (1988), Caplinger (1986)	1, 3-22
+ Frei & Steiger (1992) or Salton (1988)	1, 3-22
+ Botafogo (1992), Alain (1992)	1, 3-22
+ Shepherd (1990), Guinan (1990)	1, 3-22
Betrabet (1993) Betrabet Thesis (1993) Berk (1991) Baase (1988), Burt (1991), UCINET + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Betrabet (1993) Betrabet Thesis (1993) Berk (1991) Baase (1988) Shepherd (1990)	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
Kommers (1990) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	
+ Frisse 1988 or Frei & Steiger (1992)	1, 3-22
Dunlop (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	5-10, 12-21
Dunlop (1991) Frei & Steiger (1992) Frei & Steiger (1995) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Shepherd (1990)	1, 3-22
+ Baase (1988)	1, 3-22
Kommers (1990) , + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ Burt (1991), UCINET, Botafogo (1992) Conklin (1988), Caplinger (1986),	1, 3-22
+ Baase (1988)	1, 3-22
Croft & Turtle (1989) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Alain (1992) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Salton & McGill (1983) Salton (1971), Salton (1988), SMART + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
US 5,838,906, or Mauldin US 5,748,954	
Fox (1988) Fox Thesis (1983) Fox Collections (1983) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Fox Thesis (1983) Fox Collections (1983) Berk (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Brodda (1967) Frisse (1988) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Brodda (1967) Belew (1986) Rose (1991) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Brodda (1967) Kaplan 891 Patent + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Weiss (1996) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ NetCarta, Conklin (1988), Caplinger (1986)	1, 3-22
+ Baase (1988), Botafogo (1993)	1, 3-22
+ Pirolli (1996)	1, 3-22
Salton (1963) Pinski (1976), LATimes, Caplinger (1986), + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle	1, 3-22

Combination	Claims of the ’571 Patent Rendered Obvious by the Combination
US 5,838,906, or Mauldin US 5,748,954	
Salton & McGill (1983) Tapper (1982) , LATimes, Caplinger (1986) + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
Crouch (1989) Shepherd (1990), Caplinger (1986), + Conklin (1987), Berners-Lee (1989), Kaplan (1995), Pinkerton (1994), Doyle US 5,838,906, or Mauldin US 5,748,954	1, 3-22
+ SMART, Fox Thesis	1, 3-22
Joachims (1995), Crouch (1989), Caplinger (1986) + Pinkerton (1994), Doyle US 5,838,906, Conklin (1988), Mauldin US 5,748,954,	1, 3-22

In addition, each claim is obvious in view of cited references in combination with the general knowledge in the art. Knowledge and use of the internet is exemplified by at least the following references: Conklin, 1987; Berners-Lee, 1989; Krol, 1994; Pinkerton, 1994; LA Times; Doyle U.S. 5,838,906, Maudlin, Mauldin US 5,748,954, Schatz 1994, and Nielson 1990b.

Additional specific combinations are described in Ex. I-2, I-3, I-4, I-5, I-6, and I-7 of Table 11 in the Invalidity Contentions of January 23, 2009.

With respect to the references and combinations disclosed herein, Defendants incorporate by reference Section V.C.1 (except for Exhibit I-1) of their Invalidity Contentions of January 23, 2009. Defendants further reference the following:

Ex G-80	Brodda 1967
Ex G-81	Baase 1988
Ex G-84	Caplinger 1986

2. Motivation to Combine

With respect to the references and combinations disclosed herein, Defendants incorporate by reference Section V.C.2 of their Invalidity Contentions of January 23, 2009.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352

BASED ON BENNY BRODDA, HANS KARLGREN, "CITATION INDEX AND MEASURES OF ASSOCIATION IN MECHANIZED DOCUMENT RETRIEVAL," KVAL PM 295 (1967). REPORT NO. 2 TO THE ROYAL TREASURY. PUBLISHED BY SPRAKFORLAGET SKRIPTOR. ("BRODDA & KARLGREN, 1967")

Claim Text from '352 Patent	Brodda & Karlgren, 1967
26. A non-semantical method for numerically representing objects in a computer database and for computerized searching of the numerically represented objects in the database, wherein direct and indirect relationships exist between objects in the database, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4, 6
[26a] marking objects in the database so that each marked object may be individually identified by a computerized search;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 6
[26b] creating a first numerical representation for each identified object in the database based upon the object's direct relationship with other objects in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[26c] storing the first numerical representations for use in computerized searching;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[26d] analyzing the first numerical representations for indirect relationships existing between or among objects in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 3-4
[26e] generating a second numerical representation of each object based on the analysis of the first numerical representation;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8, 9-13, passim
[26f] storing the second numerical representation for use in computerized searching; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3-4

Claim Text from ’352 Patent	Brodda & Karlgren, 1967
[26g] searching the objects in the database using a computer and the stored second numerical representations, wherein the search identifies one or more of the objects in the database.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
27. The non-semantical method of claim 26, wherein the objects in the database include words, and semantic indexing techniques are used in combination with the non-semantical method, the method further comprising the step of creating and storing a Boolean word index for the words of the objects in the database.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4 n.1
28. The non-semantical method of claim 26 wherein the first and second numerical representations are vectors that are arranged in first and second matrices;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 9-10
[28a] the direct relationships are express references from a one object to another object in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4, 6
[28b] the objects in the database are assigned chronological data;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
[28c] and wherein the step of searching comprises the steps of matrix searching of the second matrices;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4, 5, 9-13, passim
[28d] and examining the chronological data.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 8
29. The non-semantical method of claim 26 wherein the step of analyzing the first numerical representation further comprises: examining the first numerical representation for patterns which indicate the indirect relationships.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4
30. The non-semantical method of claim 29, given	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4

Claim Text from ’352 Patent	Brodda & Karlgren, 1967
that object A occurs before object B and object c occurs before object A, and wherein the step of creating a first numerical representation comprises examining for the direct relationship B cites A and wherein the step of examining for patterns further comprises the step of examining for the following pattern: A cites c, and B cites c.	
31. The non-semantical method of claim 29, wherein a, b, c, A, d, e, f, B, g, h, and i are objects in the database and given that; a, b, and c occur before A; . . .	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4
32. The non-semantical method of claim 26, wherein the step of analyzing further comprises the step of weighing, wherein some indirect relationships are weighed more heavily than other indirect relationships.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4-5
33. The non-semantical method of claim 26, wherein the step of analyzing the first numerical representations for indirect relationships further comprises:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3-4
[33a] creating an interim vector representing each object; and wherein the step of generating a second numerical representation uses coefficients of similarity and further comprises:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 9-10
[33b] calculating Euclidean distances between interim vector representations of each object;	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[33c] creating proximity vectors representing the objects using the calculated Euclidean distances; and	

Claim Text from ’352 Patent	Brodda & Karlgren, 1967
[33d] using the proximity vectors and using coefficients of similarity to calculate the second numerical representations.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
34. The non-semantical method of claim 26, wherein objects in the database may be divided into subsets and wherein the marking step includes the step of marking subsets of objects in the database and wherein relationships exist between or among subsets of objects in the database.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 6 Further, disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
36. The non-semantical method of claim 26, wherein the step of searching the objects comprises the steps of: selecting an object; using the second numerical representation to search for objects similar to the selected object.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
37. The non-semantical method of claim 26, wherein the step of searching includes the step of graphically displaying one or more of the identified objects.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 5
38. The non-semantical method of claim 26, wherein the step of searching includes the step of identifying a paradigm object.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
39. The non-semantical method of claim 26, wherein the step of searching the objects comprises the steps of: selecting a pool of objects;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[39a] pool-similarity searching to identify a similar pool of textual objects, similar in relation to the objects in marked pool; and	*See, e.g.,* Brodda & Karlgren, 1967, at p. 2 Further, disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.

Claim Text from ’352 Patent	Brodda & Karlgren, 1967
[39b] pool-importance searching to identify an important pool of textual objects, important in relation to the objects in the selected pool.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
40. The non-semantical method of claim 26, the step of searching comprising the steps of: identifying a paradigm pool of objects; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[40a] searching for relationships between the objects and the paradigm pool of objects;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 2
[40b] wherein the searched for relationship is pool importance or pool similarity.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 2
41. A method for the non-semantical indexing of objects stored in a computer database, the method for use in searching the database for the objects, comprising the steps of: extracting, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-3
[41a] labeling objects with a first numerical representation; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 6
[41b] generating a second numerical representation for each object based on each object’s references to other objects;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[41c] patterning, comprising the step of creating a third numerical representation for each object using the second numerical representations, wherein the third numerical representation for each object is determined from an examination of the second numerical representations for occurrences of patterns that define indirect relations between or among objects;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5
[41d] weaving, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4

Claim Text from '352 Patent	Brodda & Karlgren, 1967
calculating a fourth numerical representation for each object based on the euclidean distances between the third numerical representations; and	Further, disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[41e] determining a fifth numerical representation for each object by processing the fourth numerical representations through similarity processing; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
[41f] storing the fifth numerical representations in the computer database as the index for use in searching for objects in the database.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-2
42. The method of claim 41 wherein the first through fifth numerical representations are vector representations and further comprises the step of clustering objects having similar characteristics.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 9-10
44. The method of claim 41 wherein the step of creating the third numerical representations further comprises the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5
[44a] analyzing the second numerical representation against a plurality of empirically defined patterns, wherein certain patterns are more important than others; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 4
[44b] weighing the analyzed second numerical representations according to the importance of the patterns.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
45. A method for searching indexed objects, wherein the index is stored, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-2, 4, 6
[45a] entering search commands;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
[45b] processing the search commands with a	*See, e.g.,* Brodda & Karlgren, 1967, at p.2

Claim Text from ’352 Patent	Brodda & Karlgren, 1967
processor;	
[45c] retrieving the stored index using the processor;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
[45d] analyzing the index to identify a pool of objects, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
[45e] interpreting the processed searched commands as a selection of an object;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[45f] identifying a group of objects that have a relationship to the selected object, wherein the step of identifying comprises the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2-4
[45g] identifying objects that are referred to by the selected object; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 3
[45h] identifying objects that refer to the selected object	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 3
[45i] quantifying the relationship of the selected object to each object in the group of objects; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4, 5, 8
[45j] ranking the objects in the group of objects in accordance to the quantified relationship to the selected object; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
[45k] presenting one or more objects from the group of objects in ranked order.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352

Based on Cleveland, Donald, "An n-Dimensional Retrieval Model," J. Am. Soc. Inf. Sci., pp. 342-47 (1976) ("Cleveland, 1976")

Claim Text from '352 Patent	Cleveland, 1976
26. A non-semantical method for numerically representing objects in a computer database and for computerized searching of the numerically represented objects in the database, wherein direct and indirect relationships exist between objects in the database, comprising:	*See, e.g.,* Cleveland, 1976, at *passim*, abstract. This paper reports a technique which expands W. Goffman's Indirect Method search strategy by using means other than index terms to reflect document content. The four basic measures of document relatedness were: (1) Index terms, (2) Journals in which the documents appeared, (3) Closeness of the authors of the documents and (4) Closeness of citations. (Abstract).
[26a] Marking objects in the database so that each marked object may be individually identified by a computerized search;	*See, e.g.,* Cleveland, 1976, at *passim* and p. 344, 345. n(Ji) is the number of journals representing the journal citation profile of *Ji*. (p. 344) $\Theta(C_i)$ is the number of the number of citations representing document *Ci*. (p. 345)
[26b] creating a first numerical representation for each identified object in the database based upon the object's direct relationship with other objects in the database;	*See, e.g.,* Cleveland, 1976, at p. 344-45 The second measure was based on the journals in which the documents appeared. There were 16 different journals in the data set. Approximately 30,000 citations, all the citations for a one year period, were examined. The result was a frequency list of citations for each of the 16 journals, giving the total citations to other journals in the data set. . . . Thus, connected with each journal was its journal citation profile. (p. 344)
[26c] storing the first numerical representations for use in computerized searching;	The second measure was based on the journals in which the documents appeared. There were 16 different journals in the data set. Approximately 30,000 citations, all the citations for a one year period, were examined. The result was a frequency list of citations for each of the 16 journals, giving the total citations to other journals in the data set. . . . (p. 344)

<table>
<tr><th>Claim Text from ’352 Patent</th><th>Cleveland, 1976</th></tr>
<tr><td></td><td>Θ(Ci) is the number of the number of citations representing document Ci. (p. 345)</td></tr>
<tr><td>[26d] analyzing the first numerical representations for indirect relationships existing between or among objects in the database;</td><td rowspan="3">See, e.g., Cleveland, 1976, at p. 344 -345
Thus, connected with each journal was its journal citation profile. The measure between journal Ji and journal Jj was defined to be
$$Q_{ij} = \frac{n(J_i \Lambda J_j)}{n(J_j)} \quad ,$$
where n(Ji Λ Jj) is the number of cited journals common to the profiles of journal Jj, and journal Jj and n(Ji) is the number of journals representing the journal citation profile of Jj. . . . (p. 344)
4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345)
Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$
Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document Ci and Cj, and Θ(Ci) is the number of the number of citations representing document Ci. (p. 345)
Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346).</td></tr>
<tr><td>[26e] generating a second numerical representation of each object based on the analysis of the first numerical representation;</td></tr>
<tr><td>[26f] storing the second numerical representation for use in computerized searching; and</td></tr>
<tr><td>[26g] searching the objects in the database using a computer and the stored second numerical</td><td>See, e.g., Cleveland, 1976, at passim, p. 345-46</td></tr>
</table>

Claim Text from ’352 Patent	Cleveland, 1976
representations, wherein the search identifies one or more of the objects in the database.	Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. . . . With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between the query and each document, but is based on a conditional probability of relevance between the documents in the file. (p. 346)
27. The non-semantical method of claim 26, wherein the objects in the database include words, and semantic indexing techniques are used in combination with the non-semantical method, the method further comprising the step of creating and storing a Boolean word index for the words of the objects in the database.	*See, e.g.,* Cleveland, 1976, at p. 344 & 346 1) X-axis- Keyword co-occurrence between the documents in the file. This is the measure used in the original Indirect Method experiment and is, of course, the most obvious measure. Documents with similar index terms probably have similar information content. . . . The resulting lists of index terms were used to construct a matrix of relatedness between each pair of documents in the file. . . . (p. 344) An automatic term frequency technique was used to get the index terms measure. (p. 344)

Claim Text from ’352 Patent	Cleveland, 1976
	With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between the query and each document, but is based on a conditional probability of relevance between the documents in the file. (p. 346)
28. The non-semantical method of claim 26 wherein the first and second numerical representations are vectors that are arranged in first and second matrices;	*See, e.g.,* Cleveland, 1976, at p. 345-46 Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345) Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346)
[28a] the direct relationships are express references from a one object to another object in the database;	*See, e.g.,* Cleveland, 1976, at passim, p. 344-45

Claim Text from '352 Patent	Cleveland, 1976
	The second measure was based on the journals in which the documents appeared. There were 16 different journals in the data set. Approximately 30,000 citations, all the citations for a one year period, were examined. The result was a frequency list of citations for each of the 16 journals, giving the total citations to other journals in the data set. . . . (p. 344) Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345)
[28b] the objects in the database are assigned chronological data;	*See, e.g.,* Cleveland, 1976, at p. 344 Approximately 30,000 citations, all the citations for a one year period, were examined. (p. 344)
[28c] and wherein the step of searching comprises the steps of matrix searching of the second matrices;	*See, e.g.,* Cleveland, 1976, at p. passim, 345-46 Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345) Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: . . . (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of

Claim Text from '352 Patent	Cleveland, 1976
	documents in each dimension. Eleven matrices resulted. . . . With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between the query and each document, but is based on a conditional probability of relevance between the documents in the file. (p. 346)
[28d] and examining the chronological data.	*See, e.g.,* Cleveland, 1976, at p. 344 Approximately 30,000 citations, all the citations for a one year period, were examined. (p. 344)
29. The non-semantical method of claim 26 wherein the step of analyzing the first numerical representation further comprises: examining the first numerical representation for patterns which indicate the indirect relationships.	*See, e.g.,* Cleveland, 1976, at p. 344 -345 4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345) $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Finally, the measure of citations was calculated as follows: Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345)
30. The non-semantical method of claim 29, given that object A occurs before object B and object c occurs before object A, and wherein the step of creating a first numerical representation comprises examining for the direct relationship B cites A and wherein the step of examining for patterns further comprises the step of examining for the following pattern: A cites c, and B cites c.	*See, e.g.,* Cleveland, 1976, at p. 344 -345 4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345) $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Finally, the measure of citations was calculated as follows: Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345)

Claim Text from '352 Patent	Cleveland, 1976
31. The non-semantical method of claim 29, wherein a, b, c, A, d, e, f, B, g, h, and i are objects in the database and given that; a, b, and c occur before A; A occurs before d, e, and f, which occur before B; and B occurs before g, h, and i; and wherein the step of examining for patterns further comprises the step of examining for one or more of the following patterns: (i) g cites A, and g cites B; (ii) B cites f, and f cites A; (iii) B cites f, f cites e, and e cites A; (iv) B cites f, f cites e, e cites d, and d cites A; (v) g cites A, h cites B, g cites a, and h cites a; (vi) i cites B, i cites f (or g), and f (or g) cites A; (vii) i cites g, i cites A, and g cites B; (viii) i cites g (or d), i cites h, g (or d) cites A, and h cites B; (ix) i cites a, i cites B, and A cites a; (x) i cites A, i cites e, B cites e; (xi) g cites A, g cites a, A cites a, h cites B, and h cites a; (xii) A cites a, B cites d, i cites a, and i cites d; (xiii) i cites B, i cites d, A cites a, and d cites a; (xiv) A cites b, B cites d (or c), and d (or c) cites b;	*See, e.g.,* Cleveland, 1976, at p. 344 -345 4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345) $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Finally, the measure of citations was calculated as follows: Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345)

Claim Text from '352 Patent	Cleveland, 1976
(xv) A cites b, B cites d, b cites a, and d cites a; (xvi) A cites a, B cites b, d (or c) cites a, and d (or c) cites b.	
32. The non-semantical method of claim 26, wherein the step of analyzing further comprises the step of weighing, wherein some indirect relationships are weighed more heavily than other indirect relationships.	*See, e.g.,* Cleveland, 1976, at p. 345 Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and $\Theta(Ci)$ is the number of the number of citations representing document *Ci*. (p. 345)
33. The non-semantical method of claim 26, wherein the step of analyzing the first numerical representations for indirect relationships further comprises:	(See claim 26 and below)
[33a] creating an interim vector representing each object; and wherein the step of generating a second numerical representation uses coefficients of similarity and further comprises:	*See, e.g.,* Cleveland, 1976, at p. 345-46 An automatic word frequency technique was used to get the index terms measure. This technique has been used successfully in documentation studies at Case Western Reserve University for several years. Its basic form is described by Goffman (4).

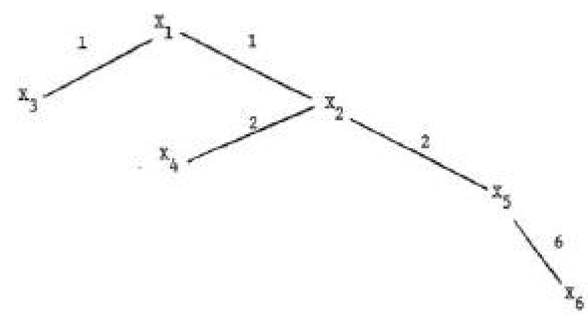
Claim Text from '352 Patent	Cleveland, 1976
	The resulting lists of index terms were used to construct a matrix of relatedness between each pair of documents in the file. The numerical value was calculated as follows: $$p_{ij} = \frac{m(X_i \Lambda X_j)}{m(X_i)}$$ where m(Xi Λ Xi) is the number of index terms common to document Xi and document Xj. m(Xi) is the total number of index terms for document Xi. (p. 344) Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of citations representing document *Ci*. (p. 345) Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346)
[33b] calculating Euclidean distances between interim vector representations of each object;	*See, e.g.,* Cleveland, 1976, at p. 345-46

Claim Text from ’352 Patent	Cleveland, 1976
	Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: . . . (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346)
[33c] creating proximity vectors representing the objects using the calculated Euclidean distances; and	*See, e.g.,* Cleveland, 1976, at p. 345-46 Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: Step One: Arbitrary thresholds were picked for each matrix in terms of the calculated numerical values. In actual practice the thresholds would depend on whether a fine or a broad scope of retrieval is desired. For pur-poses of experimentation. it is only necessary that the thresholds be held constant throughout the experiment. The thresholds picked were .14 for the index terms. .50 for thejournals, .08fortheauthorsand.0Iforthecita-tions. Any relatedness values that fell below these thres-holds were considered zero. Step Two: Go along the row of document Xi and assign a unit distance of one to each

Claim Text from '352 Patent	Cleveland, 1976
	document Xj which is above the threshold. Step Three : For each document Xj that is a distance of one from document Xi , go along the row of Xj and assign a distance of two to each document that is above the threshold, provided it is not already of distance one from document Xi. Step Four: Continue this procedure until all docu-ments have a distance from document Xi. Those documents with zero relatedness values are considered to be of infinite distance. Step Five: Repeat the procedure for all i. Step Six : Repeat the total procedure for all n basic matrices. The results are links of documents for each n basic matrix. These sequences reflect the smallest communication chain between elements, hence a quasi-distance. (p. 345)
[33d] using the proximity vectors and using coefficients of similarity to calculate the second numerical representations.	*See, e.g.,* Cleveland, 1976, at p. 344. An automatic word frequency technique was used to get the index terms measure. This technique has been used successfully in documentation studies at Case Western Reserve University for several years. Its basic form is described by Goffman (4). The resulting lists of index terms were used to construct a matrix of relatedness between each pair of documents in the file. The numerical value was calculated as follows: $p_{ij} = \frac{m(X_i \Lambda X_j)}{m(X_i)}$ where m(Xi Λ Xi) is the number of index terms common to document Xi and document Xj. m(Xi) is the total number of index terms for document Xi. (p. 344)

Claim Text from '352 Patent	Cleveland, 1976
	Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346)
34. The non-semantical method of claim 26, wherein objects in the database may be divided into subsets and wherein the marking step includes the step of marking subsets of objects in the database and wherein relationships exist between or among subsets of objects in the database.	*See, e.g.,* Cleveland, 1976, at p. 344. There were 16 different journals in the data set. Approximately 30,000 citations, all the citations for a one year period, were examined. The result was a frequency list of citations for each of the 16 journals, giving the total citations to other journals in the data set. . . . Thus, connected with each journal was its journal citation profile. (p. 344)
35. The non-semantical method of claim 34 wherein the objects are textual objects with paragraphs and the subsets are the paragraphs of the textual objects, the method further comprising the steps of:	*See, e.g.,* Cleveland, 1976, at p. 347. Cleveland also teaches applying the method to abstracts. ("[A]nother dimension, based on abstracts . . . might be used. . . .") (p. 347)
[35a] creating a subset numerical representation for each subset based upon the relationships between or among subsets;	*See, e.g.,* Cleveland, 1976, at p. 345. 4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345)
[35b] analyzing the subset numerical representations;	4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345)

Claim Text from ’352 Patent	Cleveland, 1976
	Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and $\Theta(Ci)$ is the number of the number of citations representing document *Ci*. (p. 345)
[35c] clustering the subsets into sections based upon the subset analysis; and	*See, e.g.,* Cleveland, 1976, at p. 346.
[35d] generating a section numerical representation for each section, wherein the section numerical representations are available for searching.	Since a quasi-metric space exited, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. (p. 346).
36. The non-semantical method of claim 26, wherein the step of searching the objects comprises the steps of: selecting an object; using the second numerical representation to search for objects similar to the selected object.	*See, e.g.,* Cleveland, 1976, at passim, p. 345-46 Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. . . . With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between the query and each document, but

Claim Text from '352 Patent	Cleveland, 1976
	is based on a conditional probability of relevance between the documents in the file. (p. 346)
37. The non-semantical method of claim 26, wherein the step of searching includes the step of graphically displaying one or more of the identified objects.	*See, e.g.,* Cleveland, 1976, at p. 345-46 Graphically it looks like this  (p. 345-46).
40. The non-semantical method of claim 26, the step of searching comprising the steps of: identifying a paradigm pool of objects; and	*See, e.g.,* Cleveland, 1976, at p. 346. Since a quasi-metric space exited, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. (p. 346).
[40a] searching for relationships between the objects and the paradigm pool of objects;	
[40b] wherein the searched for relationship is pool importance or pool similarity.	
41. A method for the non-semantical indexing of objects stored in a computer database, the method for use in searching the database for the objects, comprising the steps of: extracting, comprising the steps of:	*See, e.g.,* Cleveland, 1976, at passim, abstract This paper reports a technique which expands W. Goffman's Indirect Method search strategy by using means other than index terms to reflect document content. The four basic measures of document relatedness were: (1) Index terms, (2) Journals in which the documents

Claim Text from ’352 Patent	Cleveland, 1976
	appeared, (3) Closeness of the authors of the documents and (4) Closeness of citations.
[41a] labeling objects with a first numerical representation; and	*See, e.g.,* Cleveland, 1976, at passim and p. 345. $\Theta(C_i)$ is the number of the number of citations representing document *Ci*. (p. 345)
[41b] generating a second numerical representation for each object based on each object’s references to other objects;	*See, e.g.,* Cleveland, 1976, at p. 344-45 The second measure was based on the journals in which the documents appeared. There were 16 different journals in the data set. Approximately 30,000 citations, all the citations for a one year period, were examined. The result was a frequency list of citations for each of the 16 journals, giving the total citations to other journals in the data set. . . . (p. 344) $\Theta(Ci)$ is the number of the number of citations representing document *Ci*. (p. 345)
[41c] patterning, comprising the step of creating a third numerical representation for each object using the second numerical representations, wherein the third numerical representation for each object is determined from an examination of the second numerical representations for occurrences of patterns that define indirect relations between or among objects;	*See, e.g.,* Cleveland, 1976, at p. 344 -345 4) W-axis-The commonality of citations between the documents. It is assumed that closely related documents will have closely related citations. (p. 345) Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and $\Theta(Ci)$ is the number of the number of citations representing document *Ci*. (p. 345)
[41d] weaving, comprising the steps of: calculating a fourth numerical representation for	*See, e.g.,* Cleveland, 1976, at passim, p. 345-46 Distance Matrices

Claim Text from ’352 Patent	Cleveland, 1976
each object based on the euclidean distances between the third numerical representations; and	At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. . . . With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between the query and each document, but is based on a conditional probability of relevance between the documents in the file. (p. 346)
[41e] determining a fifth numerical representation for each object by processing the fourth numerical representations through similarity processing; and	*See, e.g.,* Cleveland, 1976, at p. 345-46 Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and $\Theta(Ci)$ is the number of the number of citations representing document *Ci*. (p. 345) Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being

Claim Text from ’352 Patent	Cleveland, 1976
	unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346)
[41f] storing the fifth numerical representations in the computer database as the index for use in searching for objects in the database.	*See, e.g.,* Cleveland, 1976, at Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: (p. 345) Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. . . . Therefore, a test consists of presenting queries to the system, using a particular relatedness measure or a particular combination of measures and observing how close the retrieval results approach the ideal. With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between

Claim Text from '352 Patent	Cleveland, 1976
	the query and each document, but is based on a conditional probability of relevance between the documents in the file. (p. 346)
42. The method of claim 41 wherein the first through fifth numerical representations are vector representations and further comprises the step of clustering objects having similar characteristics.	*See, e.g.,* Cleveland, 1976, at p. 346. Thus, four basic distance measures were created, representing the four basic measures under consideration. Since a quasi-metric space existed, the objective now was to combine these orthogonal measures into various one, two, three and four-dimensional measures, using the Euclidean distance formula to determine the shortest chain between neighborhoods of documents in each dimension. Eleven matrices resulted. (p. 346)
44. The method of claim 41 wherein the step of creating the third numerical representations further comprises the steps of:	*See, e.g.,* Cleveland, 1976, at p. 345 Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345)
[44a] analyzing the second numerical representation against a plurality of empirically defined patterns, wherein certain patterns are more important than others; and	
[44b] weighing the analyzed second numerical representations according to the importance of the patterns.	
45. A method for searching indexed objects, wherein the index is stored, comprising the steps of:	*See, e.g.,* Cleveland, 1976, at *passim*, abstract and p. 344 The four basic measures of document relatedness were: (1) Index terms, (2) Journals in which the documents appeared, (3) Closeness of the authors of the documents and (4) Closeness of citations. (Abstract). 1) X-axis- Keyword co-occurrence between the documents in the file. This is the measure

Claim Text from '352 Patent	Cleveland, 1976
	used in the original Indirect Method experiment and is, of course, the most obvious measure. Documents with similar index terms probably have similar information content. . . . The resulting lists of index terms were used to construct a matrix of relatedness between each pair of documents in the file. . . . (p. 344) An automatic term frequency technique was used to get the index terms measure. (p. 344)
[45a] entering search commands;	*See, e.g.,* Cleveland, 1976, at p. 346 [A] test consists of presenting queries to the system. . . . With the "Indirect Method," the query simply serves as an entry point to the file. (p. 346)
[45b] processing the search commands with a processor;	*See, e.g.,* Cleveland, 1976, at p. 346 [A] test consists of presenting queries to the system. . . . Some form of Boolean operation is the most basic of techniques. (In the experiment reported here the index terms used to represent the "query" article made up the search vectors.) For a Boolean search a query is compared with each document in the file using any Boolean operation desired. . . . With the "Indirect Method," the query simply serves as an entry point to the file. (p. 346)
[45c] retrieving the stored index using the processor;	*See, e.g.,* Cleveland, 1976, at p. 344, 345-46 An automatic term frequency technique was used to get the index terms measure. (p. 344) Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \wedge C_j)}{\Theta(C_i)}$

Claim Text from ’352 Patent	Cleveland, 1976
	Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345) • Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices: [A] test consists of presenting queries to the system. . . . Some form of Boolean operation is the most basic of techniques. (In the experiment reported here the index terms used to represent the "query" article made up the search vectors.) For a Boolean search a query is compared with each document in the file using any Boolean operation desired. . . . With the "Indirect Method," the query simply serves as an entry point to the file. (p. 346)
[45d] Analyzing the index to identify a pool of objects, comprising the steps of:	See steps below:
[45e] interpreting the processed searched commands as a selection of an object;	*See, e.g.,* Cleveland, 1976, at p. 346 [A] test consists of presenting queries to the system. . . . With the "Indirect Method," the query simply serves as an entry point to the file. (p. 346)
[45f] identifying a group of objects that have a relationship to the selected object, wherein the step	

Claim Text from '352 Patent	Cleveland, 1976
of identifying comprises the steps of:	See steps below:
[45g] Identifying objects that are referred to by the selected object; and [45h] Identifying objects that refer to the selected object	*See, e.g.,* Cleveland, 1976, at p. 345-6 Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345) • Distance Matrices At this point, the four matrices showed the relatedness between each pair of documents in terms of the four basic measures with values between 0 and 1. It was now necessary to convert these matrices into distance matrices and combine them, using the Euclidean distance formula. If the measure value between document Xi and Xi was greater than some chosen threshold, then the distance between the pair was defined as being unit distance one. The following tactic was employed to convert each of the four basic matrices into distance matrices:
[45i] quantifying the relationship of the selected object to each object in the group of objects; and	*See, e.g.,* Cleveland, 1976, at p. 345-46 Finally, the measure of citations was calculated as follows: $S_{ij} = \frac{\Theta(C_i \Lambda C_j)}{\Theta(C_i)}$ Where $\Theta(C_i \Lambda C_j)$ is the number of citations common to document *Ci* and *Cj*, and Θ(*Ci*) is the number of the number of citations representing document *Ci*. (p. 345) The results are links of documents for each n basic matrix. These sequences reflect the smallest communication chain between elements, hence a quasi-distance. Graphically, it

Claim Text from '352 Patent	Cleveland, 1976
	looks like this: . . . Thus, four basic distance measures were created. (p. 345-46)
[45j] ranking the objects in the group of objects in accordance to the quantified relationship to the selected object; and	*See, e.g.,* Cleveland, 1976, at p. 342, 346 This relevance number is a measure of the probability that the document will satisfy the request. The result of the search is an ordered list of those documents that satisfy the request, ranked according to their probable relevance. Therefore, a test consists of presenting queries to the system, using a particular relatedness measure or a particular combination of measures and observing how close the retrieval results approach the ideal. With the "Indirect Method." the query simply serves as an entry point to the file. Once a relevant document is found, the remaining retrieved documents are determined by internal file structure, independently of the query. Relevance is not a zero or one comparison between the query and each document, but is based on a conditional probability of relevance between the documents in the file. (p. 346)
[45k] presenting one or more objects from the group of objects in ranked order.	

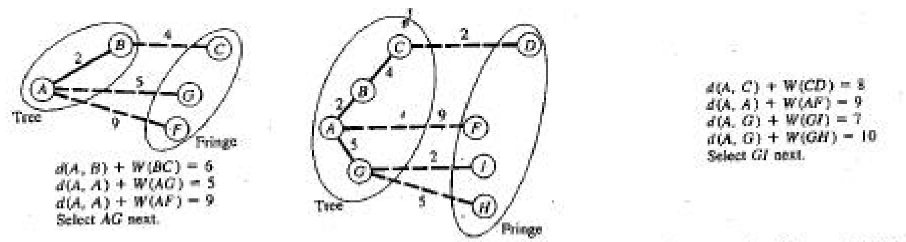
Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352

Based on Baase, S., Computer Algorithms: Introduction to Design and Analysis, 2nd Edition, Addison-Wesley Publishing Co., 1988. ("Baase, 1988)

Claim Text from '352 Patent	Baase 1988
26. A non-semantical method for numerically representing objects in a computer database and for computerized searching of the numerically represented objects in the database, wherein direct and indirect relationships exist between objects in the database, comprising:	*See, e.g.,* Baase, 1988, p. 149-156, 160-166 and 167-72, Title (Computer Algorithms).
[26a] Marking objects in the database so that each marked object may be individually identified by a computerized search;	*See, e.g.,* Baase, 1988, p. 149-156 and 167-72, Title (Computer Algorithms). Input: G = (V, E, W), a weighted graph or digraph . . . G is represented by an adjacency list structure. . . . (p. 171).
[26b] creating a first numerical representation for each identified object in the database based upon the object's direct relationship with other objects in the database;	*See, e.g.,* Baase, 1988, p. 162-163, 171 Input: G = (V, E, W), a weighted graph or digraph . . . G is represented by an adjacency list structure. . . . (p. 171).
[26c] storing the first numerical representations for use in computerized searching;	

<table>
<tr><th>Claim Text from ‘352 Patent</th><th>Baase 1988</th></tr>
<tr><td>[26d] analyzing the first numerical representations for indirect relationships existing between or among objects in the database;</td><td rowspan="3">See, e.g., Baase, 1988, at p. , 160-166, 168-172.


(b) An intermediate step. (c) An intermediate step. (CH was considered but not chosen to replace GH as a candidate.)

Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to v. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write XY. we are assuming that x is closer to v than y is. We will refer tox as tail(xy) and y as head(xy) even if G is not a directed graph.

Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge e for which d(v,tail(e))+W(e) is minimum. This is the weight of the path obtained by adjoining e to the known shortest path to tail(e).
Since the quantity d(v, tail(e))+W(e) for a candidate edge e may be used repeatedly, it can be computed once and saved. To compute it efficiently when efirst becomes a candidate, we also save d(v,y) for each y in the tree. Thus we use an array dist as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz).

After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex l and the edge Gl have just been selected. The candidate edge for F was AF, but now AF must be replaced by IF because IF yields a shorter path to F, We must also recompute
dist[F]. The vertex E, which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . .

while x ≠ w and not stuck do . . . . end { while x ≠ w and not stuck } (p. 171-172)</td></tr>
<tr><td>[26e] generating a second numerical representation of each object based on the analysis of the first numerical representation;</td></tr>
<tr><td>[26f] storing the second numerical representation for use in computerized searching; and</td></tr>
<tr><td>[26g] searching the objects in the database using a</td><td>See, e.g., Baase, 1988, at p. 149, 167, 168-172.</td></tr>
</table>

Claim Text from '352 Patent	Baase 1988
computer and the stored second numerical representations, wherein the search identifies one or more of the objects in the database.	[W]e briefly considered the problem of finding the best route between two cities on a map of airline routes. Using as our criterion the price of the plane tickets, we observed that the best – i.e., cheapest – way to get from San Diego to Sacramento was to make one stop in Los Angeles. This is one instance, or application, of a very common problem on a weighted graph or digraph: finding a shortest path between two specified vertices. The weight, or length of a path . . . in a weighted graph . . . is . . . the sum of the weights of the edges in the path. If the path is called P we denote its weight by W(P). (p. 167) {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)
28. The non-semantical method of claim 26 wherein the first and second numerical representations are vectors that are arranged in first and second matrices; the direct relationships are express references from a one object to another object in the database; the objects in the database are assigned chronological data;and wherein the step of searching comprises the steps of matrix searching of the second matrices; and examining the chronological data.	*See, e.g.,* Baase, 1988, p. 149-156, 162-163, and 167-72 Which route involves the least flying time? (p. 149).
29. The non-semantical method of claim 26 wherein the step of analyzing the first numerical representation further comprises: examining the first numerical representation for patterns which indicate the indirect relationships.	*See, e.g.,* Baase, 1988, p. 160-166, 167-72
32. The non-semantical method of claim 26, wherein the step of analyzing further comprises the step of weighing, wherein some indirect relationships are	*See, e.g.,* Baase, 1988, p. 149-156, 160-166 and 167-72

Claim Text from '352 Patent	Baase 1988
weighed more heavily than other indirect relationships.	

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352

BASED ON CROUCH, D., CROUCH, C., ANDREAS, G., "THE USE OF CLUSTER HIERARCHIES IN HYPERTEXT INFORMATION RETRIEVAL," IN HYPERTEXT '89 PROCEEDINGS, SIGCHI BULLETIN, PP. 225-237, NOVEMBER 1989. ("CROUCH, 1989")

Claim Text from '352 Patent	Crouch, 1989
26. A non-semantical method for numerically representing objects in a computer database and for computerized searching of the numerically represented objects in the database, wherein direct and indirect relationships exist between objects in the database, comprising:	*See, e.g.,* Crouch, 1989, at pp. 226, 228, 229
[26a] marking objects in the database so that each marked object may be individually identified by a computerized search;	*See, e.g.,* Crouch, 1989, at p. 230, Fig. 8
[26b] creating a first numerical representation for each identified object in the database based upon the object's direct relationship with other objects in the database;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[26c] storing the first numerical representations for use in computerized searching;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[26d] analyzing the first numerical representations for indirect relationships existing between or among objects in the database;	*See, e.g.,* Crouch, 1989, at pp. 228- 230
[26e] generating a second numerical representation of each object based on the analysis of the first numerical representation;	*See, e.g.,* Crouch, 1989, at pp. 228-230
[26f] storing the second numerical representation for use in computerized searching; and	*See, e.g.,* Crouch, 1989, at pp. 228- 230

Claim Text from ’352 Patent	Crouch, 1989
[26g] searching the objects in the database using a computer and the stored second numerical representations, wherein the search identifies one or more of the objects in the database.	*See, e.g.,* Crouch, 1989, at pp. 228, 229
27. The non-semantical method of claim 26, wherein the objects in the database include words, and semantic indexing techniques are used in combination with the non-semantical method, the method further comprising the step of creating and storing a Boolean word index for the words of the objects in the database.	*See, e.g.,* Crouch, 1989, at p. 225
28. The non-semantical method of claim 26 wherein the first and second numerical representations are vectors that are arranged in first and second matrices;	*See, e.g.,* Crouch, 1989, at p. 228
[28a] the direct relationships are express references from a one object to another object in the database;	*See, e.g.,* Crouch, 1989, at pp. 226, 228-230
[28b] the objects in the database are assigned chronological data;	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[28c] and wherein the step of searching comprises the steps of matrix searching of the second matrices;	*See, e.g.,* Crouch, 1989, at pp. 228, 229
[28d] and examining the chronological data.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.

Claim Text from ’352 Patent	Crouch, 1989
29. The non-semantical method of claim 26 wherein the step of analyzing the first numerical representation further comprises: examining the first numerical representation for patterns which indicate the indirect relationships.	*See, e.g.,* Crouch, 1989, at pp. 228- 230
30. The non-semantical method of claim 29, given that object A occurs before object B and object c occurs before object A, and wherein the step of creating a first numerical representation comprises examining for the direct relationship B cites A and wherein the step of examining for patterns further comprises the step of examining for the following pattern: A cites c, and B cites c.	*See, e.g.,* Crouch, 1989, at pp. 228- 230
31. The non-semantical method of claim 29, wherein a, b, c, A, d, e, f, B, g, h, and i are objects in the database and given that; a, b, and c occur before A; A occurs before d, e, and f, which occur before B; and B occurs before g, h, and i; and wherein the step of examining for patterns further comprises the step of examining for one or more of the following patterns: (i) g cites A, and g cites B; (ii) B cites f, and f cites A; (iii) B cites f, f cites e, and e cites A; (iv) B cites f, f cites e, e cites d, and d cites A; (v) g cites A, h cites B, g cites a, and h cites a; (vi) i cites B, i cites f (or g), and f (or g) cites A;	*See, e.g.,* Crouch, 1989, at pp. 228- 230

Claim Text from ’352 Patent	Crouch, 1989
(vii) i cites g, i cites A, and g cites B; (viii) i cites g (or d), i cites h, g (or d) cites A, and h cites B; (ix) i cites a, i cites B, and A cites a; (x) i cites A, i cites e, B cites e; (xi) g cites A, g cites a, A cites a, h cites B, and h cites a; (xii) A cites a, B cites d, i cites a, and i cites d; (xiii) i cites B, i cites d, A cites a, and d cites a; (xiv) A cites b, B cites d (or c), and d (or c) cites b; (xv) A cites b, B cites d, b cites a, and d cites a; (xvi) A cites a, B cites b, d (or c) cites a, and d (or c) cites b.	
32. The non-semantical method of claim 26, wherein the step of analyzing further comprises the step of weighing, wherein some indirect relationships are weighed more heavily than other indirect relationships.	*See, e.g.,* Crouch, 1989, at pp. 228, 229
33. The non-semantical method of claim 26, wherein the step of analyzing the first numerical representations for indirect relationships further comprises:	*See, e.g.,* Crouch, 1989, at pp. 228- 230
[33a] creating an interim vector representing each object; and wherein the step of generating a second numerical representation uses coefficients of similarity and further comprises:	*See, e.g.,* Crouch, 1989, at p. 228
[33b] calculating Euclidean distances between interim vector representations of each object;	*See, e.g.,* Crouch, 1989, at p. 228

Claim Text from '352 Patent	Crouch, 1989
[33c] creating proximity vectors representing the objects using the calculated Euclidean distances; and	*See, e.g.,* Crouch, 1989, at p. 228
[33d] using the proximity vectors and using coefficients of similarity to calculate the second numerical representations.	*See, e.g.,* Crouch, 1989, at p. 228
34. The non-semantical method of claim 26, wherein objects in the database may be divided into subsets and wherein the marking step includes the step of marking subsets of objects in the database and wherein relationships exist between or among subsets of objects in the database.	*See, e.g.,* Crouch, 1989, at p. 230, Fig. 8
35. The non-semantical method of claim 34 wherein the objects are textual objects with paragraphs and the subsets are the paragraphs of the textual objects, the method further comprising the steps of:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[35a] creating a subset numerical representation for each subset based upon the relationships between or among subsets;	
[35b] analyzing the subset numerical representations;	
[35c] clustering the subsets into sections based upon the subset analysis; and	
[35d] generating a section numerical representation for each section, wherein the section numerical representations are available for searching.	
36. The non-semantical method of claim 26, wherein the step of searching the objects comprises the steps of: selecting an object; using the second numerical representation to search for objects	*See, e.g.,* Crouch, 1989, at pp. 228, 229

Claim Text from ’352 Patent	Crouch, 1989
similar to the selected object.	
37. The non-semantical method of claim 26, wherein the step of searching includes the step of graphically displaying one or more of the identified objects.	*See, e.g.,* Crouch, 1989, at pp.226, 230
38. The non-semantical method of claim 26, wherein the step of searching includes the step of identifying a paradigm object.	*See, e.g.,* Crouch, 1989, at p. 229
39. The non-semantical method of claim 26, wherein the step of searching the objects comprises the steps of: selecting a pool of objects;	*See, e.g.,* Crouch, 1989, at p. 228
[39a] pool-similarity searching to identify a similar pool of textual objects, similar in relation to the objects in marked pool; and	*See, e.g.,* Crouch, 1989, at p. 228
[39b] pool-importance searching to identify an important pool of textual objects, important in relation to the objects in the selected pool.	*See, e.g.,* Crouch, 1989, at p. 228, 230
40. The non-semantical method of claim 26, the step of searching comprising the steps of: identifying a paradigm pool of objects; and	*See, e.g.,* Crouch, 1989, at p. 229
[40a] searching for relationships between the objects and the paradigm pool of objects;	*See, e.g.,* Crouch, 1989, at p. 229
[40b] wherein the searched for relationship is pool importance or pool similarity.	*See, e.g.,* Crouch, 1989, at p. 229
41. A method for the non-semantical indexing of objects stored in a computer database, the method for use in searching the database for the objects,	*See, e.g.,* Crouch, 1989, at pp. 225, 226, 228, 229

Claim Text from ’352 Patent	Crouch, 1989
comprising the steps of: extracting, comprising the steps of:	
[41a] labeling objects with a first numerical representation; and	*See, e.g.,* Crouch, 1989, at p. 230, Fig. 8
[41b] generating a second numerical representation for each object based on each object’s references to other objects;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[41c] patterning, comprising the step of creating a third numerical representation for each object using the second numerical representations, wherein the third numerical representation for each object is determined from an examination of the second numerical representations for occurrences of patterns that define indirect relations between or among objects;	*See, e.g.,* Crouch, 1989, at pp. 228-230
[41d] weaving, comprising the steps of: calculating a fourth numerical representation for each object based on the euclidean distances between the third numerical representations; and	*See, e.g.,* Crouch, 1989, at pp. 228, 229
[41e] determining a fifth numerical representation for each object by processing the fourth numerical representations through similarity processing; and	*See, e.g.,* Crouch, 1989, at pp. 228, 229
[41f] storing the fifth numerical representations in the computer database as the index for use in searching for objects in the database.	*See, e.g.,* Crouch, 1989, at p. 225
42. The method of claim 41 wherein the first through fifth numerical representations are vector	*See, e.g.,* Crouch, 1989, at pp. 228, 229

Claim Text from ’352 Patent	Crouch, 1989
representations and further comprises the step of clustering objects having similar characteristics.	
44. The method of claim 41 wherein the step of creating the third numerical representations further comprises the steps of:	*See, e.g.,* Crouch, 1989, at pp. 228-230
[44a] analyzing the second numerical representation against a plurality of empirically defined patterns, wherein certain patterns are more important than others; and	*See, e.g.,* Crouch, 1989, at pp. 228- 230
[44b] weighing the analyzed second numerical representations according to the importance of the patterns.	*See, e.g.,* Crouch, 1989, at p. 228
45. A method for searching indexed objects, wherein the index is stored, comprising the steps of:	*See, e.g.,* Crouch, 1989, at p. 225
[45a] entering search commands;	*See, e.g.,* Crouch, 1989, at p. 229, 230
[45b] processing the search commands with a processor;	*See, e.g.,* Crouch, 1989, at p. 229, 230
[45c] retrieving the stored index using the processor;	*See, e.g.,* Crouch, 1989, at pp. 225, 228, 229
[45d] analyzing the index to identify a pool of objects, comprising the steps of:	*See, e.g.,* Crouch, 1989, at pp. 225, 228, 229
[45e] interpreting the processed searched commands as a selection of an object;	*See, e.g.,* Crouch, 1989, at p. 228, 229
[45f] identifying a group of objects that have a	*See, e.g.,* Crouch, 1989, at pp. 228, 229, 230

Claim Text from ’352 Patent	Crouch, 1989
relationship to the selected object, wherein the step of identifying comprises the steps of:	
[45g] identifying objects that are referred to by the selected object; and	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[45h] identifying objects that refer to the selected object	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[45i] quantifying the relationship of the selected object to each object in the group of objects; and	*See, e.g.,* Crouch, 1989, at p. 228-230
[45j] ranking the objects in the group of objects in accordance to the quantified relationship to the selected object; and	*See, e.g.,* Crouch, 1989, at p. 228, 230
[45k] presenting one or more objects from the group of objects in ranked order.	*See, e.g.,* Crouch, 1989, at p. 230, 234

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352

Can and Ozkarahan, "A Dynamic Cluster Maintenance System for Information Retrieval," ACM, Vol. 6, p. 123, 1987 (Can 1987)

Claim Text from '352 Patent	Can, 1987
38. The non-semantical method of claim 26, wherein the step of searching includes the step of identifying a paradigm object.	*See, e.g.,* Can 1987 at 123, 124, 129-130
39. The non-semantical method of claim 26, wherein the step of searching the objects comprises the steps of: selecting a pool of objects;	*See, e.g.,* Can 1987 at p. 124
[39a] pool-similarity searching to identify a similar pool of textual objects, similar in relation to the objects in marked pool; and	*See, e.g.,* Can 1987 at p. 123, 124
[39b] pool-importance searching to identify an important pool of textual objects, important in relation to the objects in the selected pool.	*See, e.g.,* Can 1987 at 123, 124, 129-130
40. The non-semantical method of claim 26, the step of searching comprising the steps of: identifying a paradigm pool of objects; and	*See, e.g.,* Can 1987 at p. 124
[40a] searching for relationships between the objects and the paradigm pool of objects;	*See, e.g.,* Can 1987 at 129, 130.
[40b] wherein the searched for relationship is pool importance or pool similarity.	*See, e.g.,* Can 1987 at 129-130.

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims as appropriate, for example, depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352
BASED ON GERARD SALTON AND CHRIS BUCKLEY, "AUTOMATIC TEXT STRUCTURING AND RETRIEVAL – EXPERIMENTS IN AUTOMATIC ENCYCLOPEDIA SEARCHING ("SALTON & BUCKLEY 1991")

Claim Text from '352 Patent	SALTON & BUCKLEY 1991
[28a] the direct relationships are express references from a one object to another object in the database;	*See, e.g.*, Salton & Buckley 1991 at 25 ("Three kinds of references between articles are available in the encyclopedia, consisting of 'see', 'see also', and 'qv' (quod vide) references.").
32. The non-semantical method of claim 26, wherein the step of analyzing further comprises the step of weighing, wherein some indirect relationships are weighed more heavily than other indirect relationships.	*See, e.g.*, Salton & Buckley 1991 at Tables 1-4 (disclosing different similarity weights for second-level searches).
[33a] creating an interim vector representing each object; and wherein the step of generating a second numerical representation uses coefficients of similarity and further comprises:	*See, e.g.*, Salton & Buckley 1991 at 23 ("A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment. These term vectors form the basis for the text comparison operations.").
34. The non-semantical method of claim 26, wherein objects in the database may be divided into subsets and wherein the marking step includes the step of marking subsets of objects in the database and wherein relationships exist between or among subsets of objects in the database.	*See, e.g.*, Salton & Buckley 1991 at 23 ("A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment. These term vectors form the basis for the text comparison operations. Similarities between particular text items (or between text items and information requests) are obtained by comparing the term vectors for pairs of text items at various levels of detail. When sufficient similarities are detected in both global as well as local contexts, the texts are assumed to be related.").
35. The non-semantical method of claim 34 wherein the objects are textual objects with paragraphs and the subsets are the paragraphs of the textual objects, the method further comprising the steps of:	*See, e.g.*, Salton & Buckley 1991 at 23 ("A standard indexing system is used to assign to each text unit (that is, section, *paragraph*, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment.") (emphasis added); 24 ("Larger texts are therefore most easily processed by subdividing them into shorter units before the text comparison system is used. One possibility is to define subdocuments consisting of the various subsections of text into which these long encyclopedia articles are

Claim Text from ’352 Patent	SALTON & BUCKLEY 1991
	subdivided. Each subdocument is then used as a separate query and the outputs obtained with the several subdocuments are appropriately combined.”).
[35a] creating a subset numerical representation for each subset based upon the relationships between or among subsets;	*See* preamble to Claim 35, *supra*, *see also* Salton & Buckley 1991 at 24 (“For short text excerpts, such as text sentences, a text similarity measure that depends on the proportion of matching items is not indicative of coincidence in text meaning . . . Short texts are therefore compared using an *atn* term weight, equivalent to the numerator of expression (1) without the length normalization of the denominator. The *atn* term weight ranges in size from 0 to *log N*, and the corresponding inner product text similarity depends on the number (rather than the proportion) of matching terms.”).
[35b] analyzing the subset numerical representations;	*See supra* at Claim 35[a].
[35c] clustering the subsets into sections based upon the subset analysis; and	*See, e.g.*, Salton & Buckley 1991 at Table 3; 27 (“Long query articles consisting of many text paragraphs are best broken down into more focused parts by using the individual paragraphs as subqueries, and combining the respective search results”).
[35d] generating a section numerical representation for each section, wherein the section numerical representations are available for searching.	*See, e.g.*, Salton & Buckley 1991 at Table 3; 27 (“Long query articles consisting of many text paragraphs are best broken down into more focused parts by using the individual paragraphs as subqueries, and combining the respective search results”).
37. The non-semantical method of claim 26, wherein the step of searching includes the step of graphically displaying one or more of the identified objects.	*See, e.g.*, Salton & Buckley 1991 at Tables 1-5.
45. A method for searching indexed objects, wherein the index is stored, comprising the steps of:	*See, e.g.*, Salton & Buckley 1991 at 23 (“A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment. These term vectors form the basis for the text comparison operations.”).

Claim Text from ’352 Patent	**SALTON & BUCKLEY 1991**
[45a] entering search commands;	*See*, *e.g.*, Salton & Buckley 1991 at 24 (“An automated encyclopedia search system is implemented which uses particular encyclopedia articles as search requests, and retrieves related articles in decreasing order of presumed similarity with the request articles.”).
[45b] processing the search commands with a processor;	*See* Chart for Claim 45[a], *supra*.
[45c] retrieving the stored index using the processor;	*See* Chart for Claim 45[a], *supra*.
[45d] analyzing the index to identify a pool of objects, comprising the steps of:	*See*, *e.g.*, Salton & Buckley 1991 at 25 (disclosing a pool of four documents identified by the search).
[45e] interpreting the processed searched commands as a selection of an object;	*See*, *e.g.*, Salton & Buckley 1991 at 25 (“A standard encyclopedia search for a one-paragraph query (document 114, Acacia) is illustrated . . . .”).
[45f] identifying a group of objects that have a relationship to the selected object, wherein the step of identifying comprises the steps of:	*See*, *e.g.*, Salton & Buckley 1991 at 25 (“A standard encyclopedia search for a one-paragraph query (document 114, Acacia) is illustrated in Table 1. A multi-stage search strategy is used where all articles with a global query similarity exceeding 0.20 are retrieved initially.”).
[45g] identifying objects that are referred to by the selected object; and	*See*, *e.g.*, Salton & Buckley 1991 at 25 (“Three kinds of references between articles are available in the encyclopedia, consisting of ‘see’, ‘see also,’ and ‘qv’ (quod vide) references. The reasonable assumption can be made that when one of these references is present citing article B within, or after, the text of article A, then B is relevant to query article A.”).
[45i] quantifying the relationship of the selected object to each object in the group of objects; and	*See*, *e.g.*, Salton & Buckley 1991 at Table 1 (disclosing the quantum of similarity between the selected object (“Acacia”) and the retrieved objects (“Mimosa” and “Indigo Plant”)).
[45j] ranking the objects in the group of objects in accordance to the quantified relationship to the selected object; and	*See*, *e.g.*, Salton & Buckley 1991 at 25 (“Five items are retrieved . . . Document 23149 is retrieved at the top of the ranked list with a query similarity of 0.5058.”).

Claim Text from ’352 Patent	SALTON & BUCKLEY 1991
[45k] presenting one or more objects from the group of objects in ranked order.	*See* Chart for Claim 45[k], *supra*.

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352

Salton & Buckley, 1990, "Approaches to Text Retrieval for Structured Documents" TR 90-1083. (Department of Computer Science, Cornell University).

Claim Text from '352 Patent	Salton, 1990
34. The non-semantical method of claim 26, wherein objects in the database may be divided into subsets	*See, e.g.,* Salton, 1990, passim, e.g., p. 2-3, Fig. 2
[34a] and wherein the marking step includes the step of marking subsets of objects in the database	*See, e.g.,* Salton, 1990, p. 3, 5, 6, 11, Fig. 1 & 2.
[34b] and wherein relationships exist between or among subsets of objects in the database.	*See, e.g.,* Salton, 1990, p. 3-4, 5-6
35. The non-semantical method of claim 34 wherein the objects are textual objects with paragraphs and the subsets are the paragraphs of the textual objects, the method further comprising the steps of:	*See, e.g.,* Salton, 1990, passim and p. 2-3.
[35a] creating a subset numerical representation for each subset based upon the relationships between or among subsets;	*See, e.g.,* Salton, 1990, (p. 3-4).
[35b] analyzing the subset numerical representations;	*See, e.g.,* Salton, 1990, (p. 3-4).
[35c] clustering the subsets into sections based upon the subset analysis; and	
[35d] generating a section numerical representation for each section, wherein the section numerical representations are available for searching.	*See, e.g.,* Salton, 1990, (p. 3-4).
38. The non-semantical method of claim 26, wherein the step of searching includes the step of identifying a paradigm object.	*See, e.g.,* Salton, 1990, (p. 5).
39. The non-semantical method of claim 26, wherein the step of searching the objects comprises	*See, e.g.,* Salton, 1990, (p. 4-5).

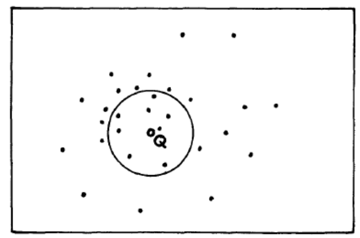
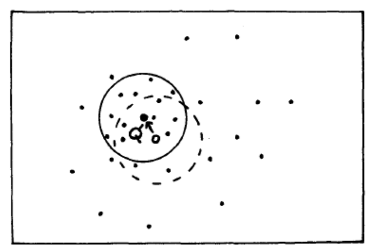
Claim Text from ’352 Patent	Salton, 1990
the steps of: selecting a pool of objects;	
[39a] pool-similarity searching to identify a similar pool of textual objects, similar in relation to the objects in marked pool; and	*See, e.g.,* Salton, 1990, (p. 4-5).
[39b] pool-importance searching to identify an important pool of textual objects, important in relation to the objects in the selected pool.	*See, e.g.,* Salton, 1990, (p. 4-5).
40. The non-semantical method of claim 26, the step of searching comprising the steps of: identifying a paradigm pool of objects; and	*See, e.g.,* Salton, 1990, (p. 4-5).
[40a] searching for relationships between the objects and the paradigm pool of objects;	*See, e.g.,* Salton, 1990, (p. 4-5).
[40b] wherein the searched for relationship is pool importance or pool similarity.	*See, e.g.,* Salton, 1990, (p. 4-5).
42. The method of claim 41 wherein the first through fifth numerical representations are vector representations and further comprises the step of clustering objects having similar characteristics.	*See, e.g.,* Salton, 1990, (p. 3-4).
45. A method for searching indexed objects, wherein the index is stored, comprising the steps of:	*See, e.g.,* Salton, 1990, passim, e.g., p. 6.
[45a] entering search commands;	*See, e.g.,* Salton, 1990 at p. 4, 6
[45b] processing the search commands with a processor;	*See, e.g.,* Salton, 1990 at p. 4, 6
[45c] retrieving the stored index using the processor;	
[45d] analyzing the index to identify a pool of objects, comprising the steps of:	
[45e] interpreting the processed searched commands as a selection of an object;	
[45f] identifying a group of objects that have a	*See, e.g.,* Salton, 1990 at p. 4, 6

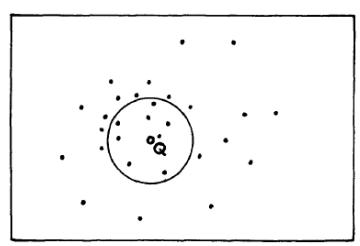
Claim Text from '352 Patent	Salton, 1990
relationship to the selected object, wherein the step of identifying comprises the steps of:	
[45g] identifying objects that are referred to by the selected object; and	*See, e.g.,* Salton, 1990 at p. 4, 5, 6
[45h] identifying objects that refer to the selected object	
[45i] quantifying the relationship of the selected object to each object in the group of objects; and	
[45j] ranking the objects in the group of objects in accordance to the quantified relationship to the selected object; and	
[45k] presenting one or more objects from the group of objects in ranked order.	

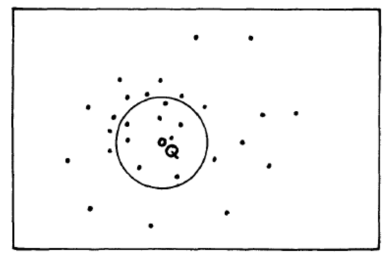
Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims as appropriate, for example, depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,544,352
BASED ON ROBERT KORFHAGE, "QUERY ENHANCEMENT BY USER PROFILES ("KORFHAGE")

Claim Text from '352 Patent	KORFHAGE
33. The non-semantical method of claim 26, wherein the step of analyzing the first numerical representations for indirect relationships further comprises:	*See infra.*
[33a] creating an interim vector representing each object; and wherein the step of generating a second numerical representation uses coefficients of similarity and further comprises:	*See* Korfhage at 112 ("We begin with the typical conceptualization of documents and queries as points in an n-dimensional document space. The classical view is that if we can suitably parameterize this space, then distance or separation between two points within the space corresponds inversely to similarity between the documents or queries that these points represent."); Figs. 1-2: **Fig. 1**  **Fig. 2** 

Claim Text from ’352 Patent	KORFHAGE
[33b] calculating Euclidean distances between interim vector representations of each object;	*See* Korfhage at 112 (“if we can suitably parameterize this space, then distance or separation between two points within the space corresponds inversely to similarity between the documents or queries that these points represent. While one can measure this distance in a number of ways, the normal measures seem to be either Euclidean or rectangular distances.”)
[33c] creating proximity vectors representing the objects using the calculated Euclidean distances; and	*See* Chart for Claims [33a] and [33b], *supra*.
[33d] using the proximity vectors and using coefficients of similarity to calculate the second numerical representations.	*See* Chart for Claims [33a] and [33b], *supra*.
41. A method for the non-semantical indexing of objects stored in a computer database, the method for use in searching the database for the objects, comprising the steps of: extracting, comprising the steps of:	*See infra*.
[41d] weaving, comprising the steps of: calculating a fourth numerical representation for each object based on the euclidean distances between the third numerical representations; and	*See* Korfhage at 112 (“if we can suitably parameterize this space, then distance or separation between two points within the space corresponds inversely to similarity between the documents or queries that these points represent. While one can measure this distance in a number of ways, the normal measures seem to be either Euclidean or rectangular distances. The former leads to retrieval (or at least examination) of all documents within an n-dimensional spherical shell around the query (Fig. 1)”); Fig. 1: **Fig. 1** 

Claim Text from '352 Patent	KORFHAGE
42. The method of claim 41 wherein the first through fifth numerical representations are vector representations and further comprises the step of clustering objects having similar characteristics.	*See* Korfhage at 112 ("("if we can suitably parameterize this space, then distance or separation between two points within the space corresponds inversely to similarity between the documents or queries that these points represent. While one can measure this distance in a number of ways, the normal measures seem to be either Euclidean or rectangular distances. *The former leads to retrieval (or at least examination) of all documents within an n-dimensional spherical shell around the query* (Fig. 1)") (emphasis added); Fig. 1: **Fig. 1** 

INVALIDITY CLAIM CHART FOR US PATENT NO. 5,832,494

BASED ON BENNY BRODDA, HANS KARLGREN, "CITATION INDEX AND MEASURES OF ASSOCIATION IN MECHANIZED DOCUMENT RETRIEVAL," KVAL PM 295 (1967). REPORT NO. 2 TO THE ROYAL TREASURY. PUBLISHED BY SPRAKFORLAGET SKRIPTOR. ("BRODDA & KARLGREN, 1967")

Claim Text from '494 Patent	Brodda & Karlgren, 1967
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4
[1a] selecting a node for analysis;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4
[1b] generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5
[1c] deriving actual cluster links from the candidate cluster links;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 5, passim, pp. 9-13.
[1d] identifying one or more nodes for display; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
[1e] displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
2. The method of claim 1 wherein each link is given a length, the step of generating the candidate cluster links comprises the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5 Further, disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.

Claim Text from ’494 Patent	Brodda & Karlgren, 1967
[2a] choosing a number as the maximum number of link lengths that will be examined; and	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[2b] examining only those links which are less than the maximum number of link lengths.	*See e.g.*, Brodda & Karlgren, 1967, at pp. 9-13.
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of: selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
5. The method of claim 1 wherein the step of generating the candidate cluster links comprises the step of: eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
9. The method of claim 8, wherein one or more nodes provide links to more than one independent application which can be executed as an extension, the method further comprising the steps of:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[9a] displaying a list of independent applications linked to the node, wherein the step of accessing accesses an independent application.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6 Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.

Claim Text from ’494 Patent	Brodda & Karlgren, 1967
12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2-4
[12a] selecting an object to determine the proximity of other objects to the selected object;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[12b] generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect relationships in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 8
[12c] deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and	*See, e.g.,* Brodda & Karlgren, 1967, at p. 5
[12d] displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
13. The method of 12 wherein a set of direct links exists for the database, and wherein the step of generating a candidate cluster link set comprises: recursively analyzing portions of the set of direct links for indirect links.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 9-13.
14. A method for representing the relationship between nodes using stored direct links, paths, and candidate cluster links, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
[14a] initializing a set of candidate cluster links;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 8
[14b] selecting the destination node of a path as the selected node to analyze;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
[14c] retrieving the set of direct links from the selected node to any other node in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 3

Claim Text from ’494 Patent	Brodda & Karlgren, 1967
[14d] determining the weight of the path using the retrieved direct links;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
[14e] repeating steps b through d for each path; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4, 5, 12-13
[14f] storing the determined weights as candidate cluster links.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4
15. The method of claim 14 further comprising the step of deriving the actual cluster links wherein the actual cluster links are a subset of the candidate cluster links.	*See, e.g.,* Brodda & Karlgren, 1967, at p. 5, 9-13.
16. The method of claim 15 wherein the step of deriving comprises the step of choosing the top rated candidate cluster links.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4
18. A method of analyzing a database having objects and a first numerical representation of direct relationships in the database, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[18a] generating a second numerical representation using the first numerical representation, wherein the second numerical representation accounts for indirect relationships in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8
[18b] storing the second numerical representation;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4
[18c] identifying at least one object in the database, wherein the stored numerical representation is used to identify objects; and	*See, e.g.,* Brodda & Karlgren, 1967, at p. 4

Claim Text from ’494 Patent	Brodda & Karlgren, 1967
[18d] displaying one or more identified objects from the database.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
19. The method of claim 18 wherein the step of generating a second numerical representation comprises: selecting an object in the database for analysis;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[19a] analyzing the direct relationships expressed by the first numerical representation for indirect relationships involving the selected object; and creating a second numerical representation of the direct and indirect relationships involving the selected object.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8
20. The method of 18 wherein the step of identifying at least one object in the database comprises: searching for objects in a database using the stored numerical representation, wherein direct and/or indirect relationships are searched.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4, 5, 8
21. The method of claim 18 wherein the displaying step comprises: generating a graphical display for representing an object in the database.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 5, 6
23. A method of representing data in a computer database with relationships, comprising the steps of:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3
[23a] assigning nodes node identifications;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 6
[23b] generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3

Claim Text from ’494 Patent	Brodda & Karlgren, 1967
[23c] allocating a weight to each link, wherein the weight signifies the strength of the relationship represented by the link relative to the strength of other relationships represented by other links; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 4, 5
[23d] displaying a node identification.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
24. The method of claim 23, wherein the data in the database is objects, wherein the nodes represent objects and each object is assigned a node identification, and wherein the relationships that exist comprise direct relationships between objects, further comprising the step of: searching generated links, wherein nodes are located by searching the generated links.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 6, 8
33. A method of representing data in a computer database and for computerized searching of the data, wherein relationships exist in the database, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3
[33a] assigning links to represent relationships in the database;	*See, e.g.,* Brodda & Karlgren, 1967, at p. 2
[33b] generating node identifications based upon the assigned links, wherein node identifications are generated so that each link represents a relationship between two identified nodes;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[33c] storing the links and node identifications, wherein the links and nodes may be retrieved;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[33d] searching for node identifications using the stored links; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8
[33e] displaying node identifications, wherein the displayed node identifications are located in the	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10

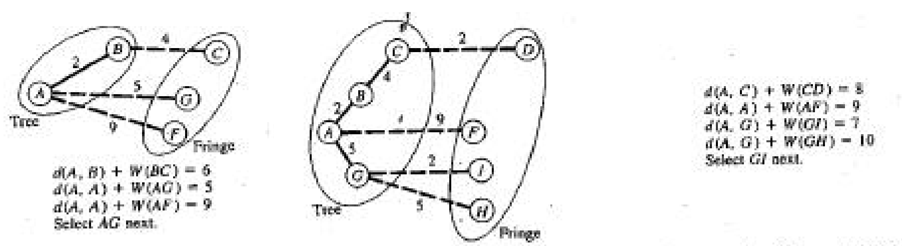
Claim Text from ’494 Patent	Brodda & Karlgren, 1967
searching step.	

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR US PATENT NO. 5,832,494

Based on Baase, S., Computer Algorithms: Introduction to Design and Analysis, 2nd Edition, Addison-Wesley Publishing Co., 1988. ("Baase, 1988)

Claim Text from '494 Patent	Baase, 1988
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.,* Baase, 1988, p. 149-156, 160-166 and 167-72, Title (Computer Algorithms). Input: G = (V, E, W), a weighted graph or digraph . . . G is represented by an adjacency list structure. . . . (p. 171).
[1a] Selecting a node for analysis;	*See, e.g.,* Baase, 1988, at p. 149-156, 160-166 and 168-172 Djisktra's shortest path algorithm will find shortest paths from v to the other vertices in order of increasing distance from v. . . . The algorithm starts at one vertex (v) and "branches out" by selecting certain edges that lead to new vertices (p. 168) x:= v (p. 171).
[1b] Generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.  (b) An intermediate step. (c) An intermediate step. (CH was considered but not chosen to replace GH as a candidate.) Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY*. we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even
[1c] Deriving actual cluster links from the candidate cluster links;	

Claim Text from '494 Patent	Baase, 1988
	if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *Gl* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recompute *dist[F]*. The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . while x ≠ w and not *stuck* do . . . . end { while x ≠ w and not *stuck* } (p. 171-172)
[1d] identifying one or more nodes for display; and	*See, e.g.,* Baase, 1988, at p. 149, 166, 167, 168-172. [W]e briefly considered the problem of finding the best route between two cities on a map of airline routes. Using as our criterion the price of the plane tickets, we observed that the best – i.e., cheapest – way to get from San Diego to Sacramento was to make one stop in Los Angeles. This is one instance, or application, of a very common problem on a weighted graph or digraph: finding a shortest path between two specified vertices. The weight, or length of a path . . . in a weighted graph . . . is . . . the sum of the weights of the edges in the path. If the path is called P we denote its weight by W(P). (p. 167) {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)

Claim Text from ’494 Patent	Baase, 1988
[1e] displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.,* Baase, 1988, at p. , 160-166, 168-172, including e.g. {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)
2. The method of claim 1 wherein each link is given a length, the step of generating the candidate cluster links comprises the steps of:	
[2a] Choosing a number as the maximum number of link lengths that will be examined; and	*See, e.g.,* Baase, 1988, at p. , 160-166, 168-172, 186-190, including e.g. If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy} parent[y] := x; dist[y] := dist[x] + ptr!.eight; end (p. 172)
[2b] examining only those links which are less than the maximum number of link lengths.	
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of: selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	*See, e.g.,* Baase, 1988, at p. , 160-166 , 168-172, 193-197 including e.g. If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy} parent[y] := x; dist[y] := dist[x] + ptr!.eight; end Traverse the fringe list to find a vertex with minimum dist; x := this vertex remove x from the fringe list status[x] := intree (p. 172)
5. The method of claim 1 wherein the step of	*See, e.g.,* Baase, 1988, at p. , 160-166, 168-172, 193-197 including e.g.

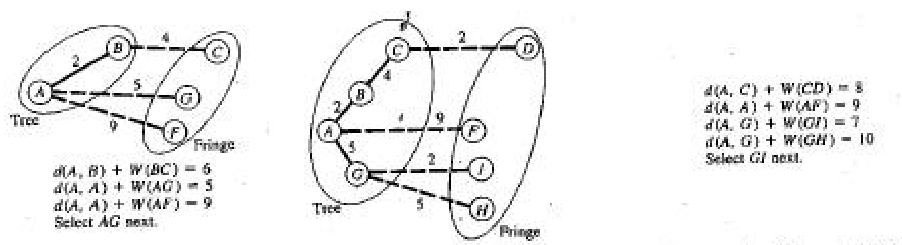
<table>
<tr><th>Claim Text from ’494 Patent</th><th>Baase, 1988</th></tr>
<tr><td>generating the candidate cluster links comprises the step of:

eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.</td><td>If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy} parent[y] := x; dist[y] := dist[x] + ptr!.eight; end

Traverse the fringe list to find a vertex with minimum dist;
x := this vertex
remove x from the fringe list
status[x] := intree (p. 172)</td></tr>
<tr><td>12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:</td><td>See p. , 160-166 , 167-72, Title (Computer Algorithms).

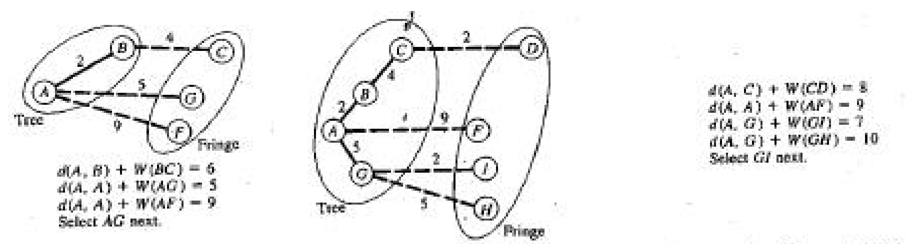
Input: G = (V, E, W), a weighted graph or digraph . . . G is represented by an adjacency list structure. . . . (p. 171).</td></tr>
<tr><td>[12a] Selecting an object to determine the proximity of other objects to the selected object;</td><td>See, e.g., Baase, 1988, at p. 160, 164-167, 168-172, 184-91, 193-97.
Djisktra’s shortest path algorithm will find shortest paths from v to the other vertices in order of increasing distance from v. . . . The algorithm starts at one vertex (v) and “branches out” by selecting certain edges that lead to new vertices (p. 168)

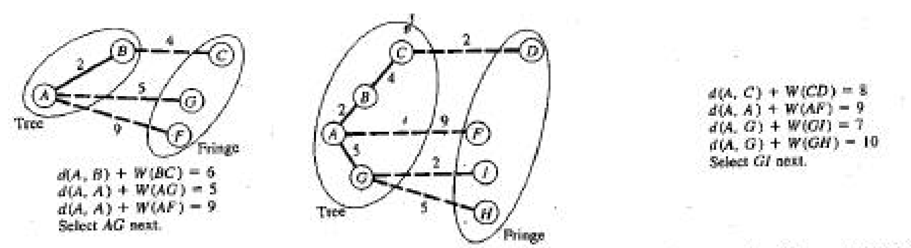
x:= v (p. 171).</td></tr>
<tr><td>[12b] generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect relationships in the database;</td><td rowspan="2">See, e.g., Baase, 1988, at p. 160-167, 168-172, 184-91, 193-97.



(b) An intermediate step.
(c) An intermediate step. (CH was considered but not chosen to replace GH as a candidate.)</td></tr>
<tr><td>[12c] Deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and</td></tr>
</table>

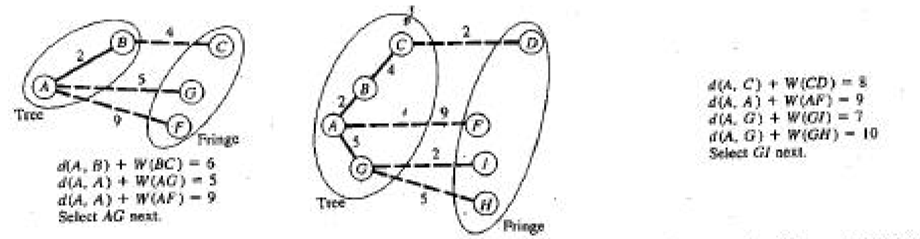
Claim Text from '494 Patent	Baase, 1988
	Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY*. we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *Gl* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recompute *dist[F].* The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . while x ≠ w and not *stuck* do . . . . end {while x ≠ w and not *stuck* } (p. 171-172)
[12d] Displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See, e.g.,* Baase, 1988, at p. 149, 166, 168-172, 184-91, 193-97including e.g. {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)
13. The method of 12 wherein a set of direct links exists for the database, and wherein the step of generating a candidate cluster link set comprises: recursively analyzing portions of the set of direct	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 184-91, 193-97. Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a

Claim Text from ’494 Patent	Baase, 1988
links for indirect links.	fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY*. we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *Gl* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recompute *dist[F].* The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . while x ≠ w and not *stuck* do . . . . end {while x ≠ w and not *stuck* } (p. 171-172)
14. A method for representing the relationship between nodes using stored direct links, paths, and candidate cluster links, comprising the steps of:	See below:
[14a] initializing a set of candidate cluster links;	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 184-91, 193-97.  (b) An intermediate step. (c) An intermediate step. (*CH* was considered but not chosen to replace *GH* as a candidate.)

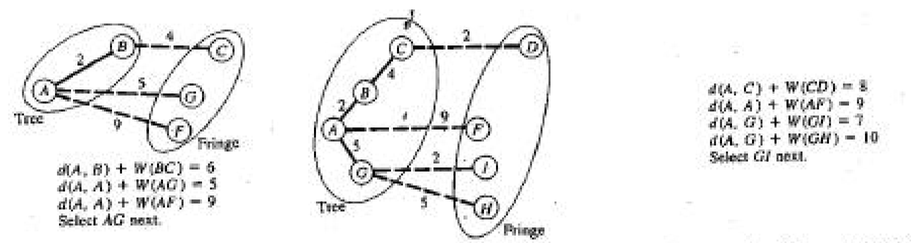
Claim Text from ’494 Patent	Baase, 1988
	Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY.* we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *GI* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recomputed *dist[F]*. The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . while x ≠ w and not *stuck* do . . . . end { x ≠ w and not *stuck* } (p. 171-172)
[14b] Selecting the destination node of a path as the selected node to analyze;	*See, e.g.,* Baase, 1988, at p. 160-161, 164-167, 168-172, 189-190, 196-197.  (b) An intermediate step. (c) An intermediate step. (*CH* was considered but not chosen to replace *GH* as a candidate.) Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write

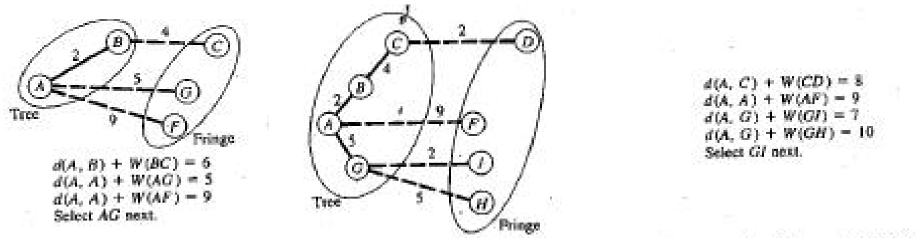
Claim Text from ’494 Patent	Baase, 1988
	XY. we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *GI* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recompute *dist[F].* The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . Traverse the fringe list to find a vertex with minimum dist; x := this vertex remove x from the fringe list status[x] := intree (p. 172)
[14c] retrieving the set of direct links from the selected node to any other node in the database;	See *supra*, including (p. 160-167, 168-172): {Traverse the adjacency list for x.} ptr:= adjacencyList[x]; while ptr ≠ nil do . . . end {while ptr ≠ nil}; . . .. end {x ≠ w and not *stuck*} (p. 171-172)
[14d] Determining the weight of the path using the retrieved direct links;	See *supra*, including (p. 160-167, 168-172):

Claim Text from ’494 Patent	Baase, 1988
	while ptr ≠ nil do . . . end {while ptr ≠ nil}; . . . If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy.} . . . If status[y] = unseen then . . . dist[y] := dist[x] + ptr!.weight . . .. end {while ptr ≠ nil}
[14e] repeating steps b through d for each path; and	See *supra*, including (p. 160-167, 168-172): while x ≠ w and not *stuck* do . . . . while ptr ≠ nil do . . . If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy.} . . . If status[y] = unseen then . . . dist[y] := dist[x] + ptr!.weight . . .. end {while ptr ≠ nil} end { x ≠ w and not *stuck* }; . . . (p. 171-172)
[14f] Storing the determined weights as candidate cluster links.	See *supra*, including (p. 160-167, 168-172): while ptr ≠ nil do . . . If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy.} Parent[y] := x; dist[y] := dist[x] + ptr!weight; end; . . If status[y] = unseen then . . . Parent[y] := x; dist[y] := dist[x] + ptr!.weight . . .. end {while ptr ≠ nil}

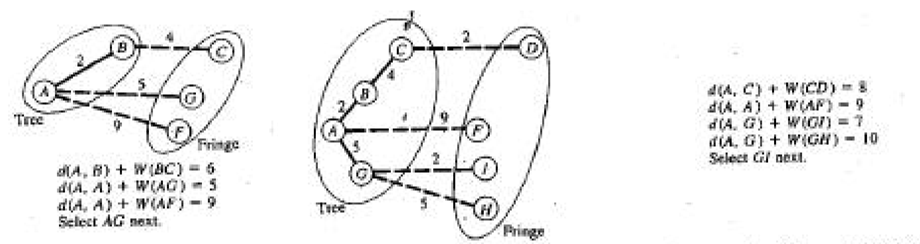
Claim Text from ’494 Patent	Baase, 1988
15. The method of claim 14 further comprising the step of deriving the actual cluster links wherein the actual cluster links are a subset of the candidate cluster links.	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 184-91, 193-97  (b) An intermediate step. (c) An intermediate step. (*CH* was considered but not chosen to replace *GH* as a candidate.) Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY.* we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *Gl* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recompute *dist[F]*. The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . Traverse the fringe list to find a vertex with minimum dist; x := this vertex remove x from the fringe list

Claim Text from ’494 Patent	Baase, 1988
	status[x] := intree (p. 172)
16. The method of claim 15 wherein the step of deriving comprises the step of choosing the top rated candidate cluster links.	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172. Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY.* we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *GI* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recomputed *dist[F].* The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . Traverse the fringe list to find a vertex with minimum dist; x := this vertex remove x from the fringe list status[x] := intree (p. 172)
18. A method of analyzing a database having objects and a first numerical representation of direct relationships in the database, comprising the steps of:	*See* p. 149-156, 160-167, 167-72, Title (Computer Algorithms). Input: G = (V, E, W), a weighted graph or digraph . . . G is represented by an adjacency list

Claim Text from ’494 Patent	Baase, 1988
	structure. . . . (p. 171).
[18a] generating a second numerical representation using the first numerical representation, wherein the second numerical representation accounts for indirect relationships in the database;	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172.  (b) An intermediate step. (c) An intermediate step. (*CH* was considered but not chosen to replace *GH* as a candidate.) Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY.* we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph. Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *Gl* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recomputed *dist[F]*. The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . while ptr ≠ nil do . . . If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by
[18b] storing the second numerical representation;	

Claim Text from ’494 Patent	Baase, 1988
	xy.} Parent[y] := x; dist[y] := dist[x] + ptr!weight; end; . . If status[y] = unseen then . . . Parent[y] := x; dist[y] := dist[x] + ptr!.weight . . .. end {while ptr ≠ nil}
[18c] identifying at least one object in the database, wherein the stored numerical representation is used to identify objects; and	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, including:  (b) An intermediate step. (c) An intermediate step. (*CH* was considered but not chosen to replace *GH* as a candidate.) while ptr ≠ nil do . . . If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y’s candidate edge by xy.} Parent[y] := x; dist[y] := dist[x] + ptr!weight; end; . . If status[y] = unseen then . . . Parent[y] := x; dist[y] := dist[x] + ptr!.weight . . ..

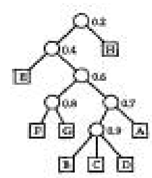
Claim Text from '494 Patent	Baase, 1988
	end {while ptr ≠ nil}
[18d] displaying one or more identified objects from the database.	*See, e.g.,* Baase, 1988, at p. 149-156, 167, and 168-172, including: {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)
19. The method of claim 18 wherein the step of generating a second numerical representation comprises: selecting an object in the database for analysis;	*See, e.g.,* Baase, 1988, at p. 160-161, 168-172, including: Djisktra's shortest path algorithm will find shortest paths from v to the other vertices in order of increasing distance from v. . . . The algorithm starts at one vertex (v) and "branches out" by selecting certain edges that lead to new vertices (p. 168) x:= v (p. 171).
[19a] analyzing the direct relationships expressed by the first numerical representation for indirect relationships involving the selected object; and creating a second numerical representation of the direct and indirect relationships involving the selected object.	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, including: If status[y] = fringe and dist[x] + ptr!.weight < dist[y] then {Replace y's candidate edge by xy.} Parent[y] := x; dist[y] := dist[x] + ptr!weight; end; . . If status[y] = unseen then . . . Parent[y] := x; dist[y] := dist[x] + ptr!.weight . . .. end {while ptr ≠ nil}

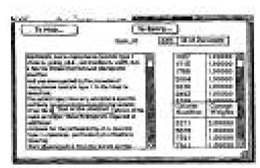
Claim Text from ’494 Patent	Baase, 1988
20. The method of 18 wherein the step of identifying at least one object in the database comprises: searching for objects in a database using the stored numerical representation, wherein direct and/or indirect relationships are searched.	*See, e.g.,* Baase, 1988, at p. 149-156, 160-167, and 168-172. [W]e briefly considered the problem of finding the best route between two cities on a map of airline routes. Using as our criterion the price of the plane tickets, we observed that the best – i.e., cheapest – way to get from San Diego to Sacramento was to make one stop in Los Angeles. This is one instance, or application, of a very common problem on a weighted graph or digraph: finding a shortest path between two specified vertices. The weight, or length of a path . . . in a weighted graph . . . is . . . the sum of the weights of the edges in the path. If the path is called P we denote its weight by W(P). (p. 167)
21. The method of claim 18 wherein the displaying step comprises: generating a graphical display for representing an object in the database.	*See, e.g.,* Baase, 1988, at p. 149-156, 160-161, 169  (b) An intermediate step. (c) An intermediate step. (CH was considered but not chosen to replace GH as a candidate.)
23. A method of representing data in a computer database with relationships, comprising the steps of: [23a] assigning nodes node identifications; [23b] generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists; [23c] allocating a weight to each link, wherein the weight signifies the strength of the relationship represented by the link relative to the strength of other relationships represented by other links; and	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172. [W]e briefly considered the problem of finding the best route between two cities on a map of airline routes. Using as our criterion the price of the plane tickets, we observed that the best – i.e., cheapest – way to get from San Diego to Sacramento was to make one stop in Los Angeles. This is one instance, or application, of a very common problem on a weighted graph or digraph: finding a shortest path between two specified vertices. The weight, or length of a path . . . in a weighted graph . . . is . . . the sum of the weights of the edges in the path. If the path is called P we denote its weight by W(P). (p. 167) Input: G = (V, E, W), a weighted graph or digraph . . . G is represented by an adjacency list structure. . . . (p. 171).

INVALIDITY CLAIM CHART FOR US PATENT NO. 5,832,494
BASED ON CROUCH, D., CROUCH, C., ANDREAS, G., "THE USE OF CLUSTER HIERARCHIES IN HYPERTEXT INFORMATION RETRIEVAL," IN HYPERTEXT '89 PROCEEDINGS, SIGCHI BULLETIN, PP. 225-237, NOVEMBER 1989. ("CROUCH, 1989")

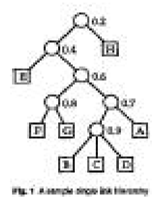
Claim Text from '494 Patent	Crouch, 1989
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.,* Crouch, 1989, at pp. 226, 228, 229 In hypertext information retrieval, each node is generally assumed to be a single document. Links exist which connect each document to other documents having keywords in common with it; the semantics of the links between nodes are keywords (document index terms) or some descriptive information representing the connected documents. In this paper we introduce an hierarchical structure which provides additional semantic information within and between nodes. This structure seems particularly well suited to the user's exploration of a document collection in a visual context. The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226) The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up the tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively in automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic

Claim Text from ’494 Patent	Crouch, 1989
	cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
selecting a node for analysis;	*See, e.g.,* Crouch, 1989, at p. 228 The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. (Crouch, 1989, p. 228)
generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally

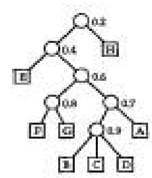
Claim Text from ’494 Patent	Crouch, 1989
	made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on

Claim Text from '494 Patent	Crouch, 1989
	the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
deriving actual cluster links from the candidate cluster links;	*See* above
identifying one or more nodes for display; and	*See, e.g.,* Crouch, 1989, at p. 234 In general, a tree representation of a clustered collection is too large to be displayed in its entirely. Therefore, a user is presented with two views of the cluster tree simultaneously: a local view containing the subtree within which the user is currently browsing (see Fig. 4) and a global view, a more comprehensive view of the tree containing a significantly larger number of nodes than the local view (see Fig. 5). A user-directed traversal among the nodes is simultaneously reflected in both displays. The global view permits the user to observe where the search is being conducted in relation to the entire tree while the loca1 view provides the user with more detailed information about a specific subtree. (Crouch, 1989, p. 230) Figure 8  (Crouch, 1989, p. 234)
displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.,* Crouch, 1989, at p. 234 In general, a tree representation of a clustered collection is too large to be displayed in its entirely. Therefore, a user is presented with two views of the cluster tree simultaneously: a

Claim Text from ’494 Patent	Crouch, 1989
	local view containing the subtree within which the user is currently browsing (see Fig. 4) and a global view, a more comprehensive view of the tree containing a significantly larger number of nodes than the local view (see Fig. 5). A user-directed traversal among the nodes is simultaneously reflected in both displays. The global view permits the user to observe where the search is being conducted in relation to the entire tree while the loca1 view provides the user with more detailed information about a specific subtree. (Crouch, 1989, p. 230) Figure 8  (Crouch, 1989, p. 234)
2. The method of claim 1 wherein each link is given a length, the step of generating the candidate cluster links comprises the steps of:	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller

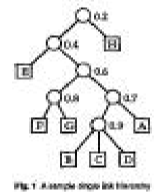
Claim Text from ’494 Patent	Crouch, 1989
	clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are

Claim Text from ’494 Patent	Crouch, 1989
	also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
choosing a number as the maximum number of link lengths that will be examined; and	*See* above and further, disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants’ P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
examining only those links which are less than the maximum number of link lengths.	*See* above.
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of: selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	*See, e.g.,* Crouch, 1989, at pp. 228, 229, 230 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the

Claim Text from ’494 Patent	Crouch, 1989
	similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high

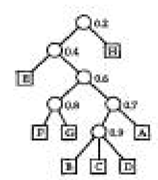
Claim Text from '494 Patent	Crouch, 1989
	kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists the value of the correlation measure of the query vector with either the centroid vector or the document vector associated with each node in the subtree. During the search process the user may change the correlation measure being calculated by means of the Correlation Measure pop-up menu. At present, the system provides a choice of several measures including vector product, inner product, Tanimoto, cosine and overlap. (Crouch, 1989, p. 230) Lists the value of the correlation measure of the query vector with either the centroid vector or the document vector associated with each node in the subtree. During the search process the user may change the correlation measure being calculated by means of the Correlation Measure pop-up menu. At present, the system provides a choice of several measures including vector product, inner product, Tanimoto, cosine and overlap. • Provides a listing of the concepts contained within the query vector (see also Fig. 6). This information is also displayed in the query window; however, in the tree display, the concepts in the query are displayed in ascending order of document frequency. The user may alter the query by adding or deleting concepts from the query vector during the search process without returning to the query window ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
5. The method of claim 1 wherein the step of generating the candidate cluster links comprises the step of:	*See, e.g.,* Crouch, 1989, at pp. 228, 229

Claim Text from '494 Patent	Crouch, 1989
eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.	Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of

Claim Text from ’494 Patent	Crouch, 1989
	association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists the value of the correlation measure of the query vector with either the centroid vector or the document vector associated with each node in the subtree. During the search process the user may change the correlation measure being calculated by means of the Correlation Measure pop-up menu. At present, the system provides a choice of several measures including vector product, inner product, Tanimoto, cosine and overlap. • Provides a listing of the concepts contained within the query vector (see also Fig. 6). This information is also displayed in the query window; however, in the tree display, the concepts in the query are displayed in ascending order of document frequency. The user may alter the query by adding or deleting concepts from the query vector during the search process without returning to the query window ...

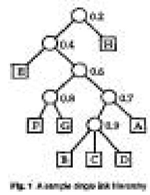
Claim Text from '494 Patent	Crouch, 1989
	Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
7. The method of claim 1, wherein one or more nodes provide external connections to objects external to the database, the method further comprising the steps of:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants' P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
activating the desired node; and	
accessing the external object linked to the node.	
8. The method of claim 7, wherein the external object is an independent application which can be executed in background, the method further comprising the step of:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants' P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
executing the independent application.	
9. The method of claim 8, wherein one or more nodes provide links to more than one independent application which can be executed as an extension, the method further comprising the steps of:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants' P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
displaying a list of independent applications linked to the node, wherein the step of accessing accesses an independent application.	*See, e.g.,* Crouch, 1989, at p. 230 Provides a listing of the concepts contained within the query vector (see also Fig. 6). This information is also displayed in the query window; however, in the tree display, the concepts in the query are displayed in ascending order of document frequency. The user may alter the query by adding or deleting concepts from the query vector during the search process without returning to the query window. • Uses different iconic representations to distinguish relevant documents from the other

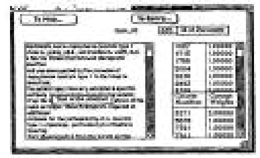
Claim Text from '494 Patent	Crouch, 1989
	documents in the tree. A list of the documents which the user has chosen as relevant to the query is maintained in the display. The user may freely insert document identifiers into and delete items from this list. The icons of the documents in this list are then highlighted in the tree representation. • Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
10. The method of claim 8, wherein the connection provides the independent application access to the information stored within the database.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants' P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
11. The method of claim 7, wherein the external connection is to another computer, wherein information is located that can be accessed, the step of accessing further comprising the step of:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants' P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
accessing the information located within the computer.	
12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject

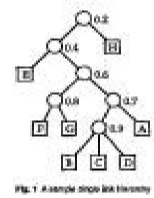
Claim Text from ’494 Patent	Crouch, 1989
	descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)  Figure 1 Fig. 1 A sample single link hierarchy (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search

Claim Text from '494 Patent	Crouch, 1989
	will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists the value of the correlation measure of the query vector with either the centroid vector or the document vector associated with each node in the subtree. During the search process the user may change the correlation measure being calculated by means of the Correlation Measure pop-up menu. At present, the system provides a choice of several measures including vector product, inner product, Tanimoto, cosine and overlap. (Crouch, 1989, p. 230)
selecting an object to determine the proximity of other objects to the selected object;	*See, e.g.,* Crouch, 1989, at p. 228 The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. (Crouch, 1989, p. 228)
generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect	*See, e.g.,* Crouch, 1989, at pp. 228, 229

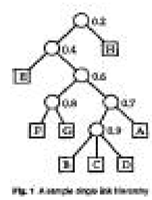
Claim Text from ’494 Patent	Crouch, 1989
relationships in the database;	Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of

Claim Text from ’494 Patent	Crouch, 1989
	association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)  Figure 1 (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and	*See* above.
displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See, e.g.,* Crouch, 1989, at p. 234

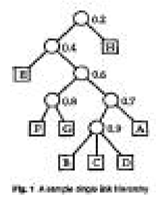
Claim Text from '494 Patent	Crouch, 1989
	 Figure 8 (Crouch, 1989, p. 234) Clicking on a terminal node (a document icon) results in the display of additional information associated with the document. (Crouch, 1989, p. 233)
13. The method of 12 wherein a set of direct links exists for the database, and wherein the step of generating a candidate cluster link set comprises: recursively analyzing portions of the set of direct links for indirect links.	*See, e.g.,* Crouch, 1989, at pp. 228-230 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the

Claim Text from ’494 Patent	Crouch, 1989
	similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch,

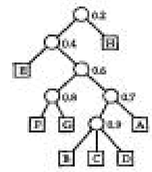
Claim Text from '494 Patent	Crouch, 1989
	1989, p. 229) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
14. A method for representing the relationship between nodes using stored direct links, paths, and candidate cluster links, comprising the steps of:	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the

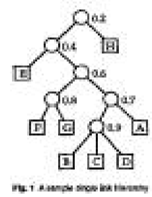
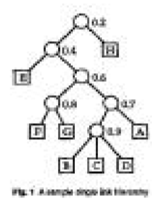
Claim Text from ’494 Patent	Crouch, 1989
	node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)

Claim Text from ’494 Patent	Crouch, 1989
initializing a set of candidate cluster links;	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the

Claim Text from ’494 Patent	Crouch, 1989
	similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
selecting the destination node of a path as the selected node to analyze;	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar

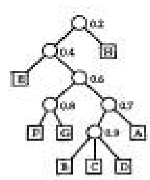
Claim Text from ’494 Patent	Crouch, 1989
	documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)

Claim Text from ’494 Patent	Crouch, 1989
	 Figure 1 (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
retrieving the set of direct links from the selected node to any other node in the database;	*See, e.g.,* Crouch, 1989, at pp. 228, 230 The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a

Claim Text from ’494 Patent	Crouch, 1989
	straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. (Crouch, 1989, p. 228) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
determining the weight of the path using the retrieved direct links;	*See, e.g.,* Crouch, 1989, at Figure 1  (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C and D in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)
repeating steps b through d for each path; and	*See, e.g.,* Crouch, 1989, at pp. 228, 229
storing the determined weights as candidate cluster links.	*See, e.g.,* Crouch, 1989, at 228. Figure 1  (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it.

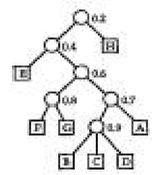
Claim Text from '494 Patent	Crouch, 1989
	The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C and D in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)
15. The method of claim 14 further comprising the step of deriving the actual cluster links wherein the actual cluster links are a subset of the candidate cluster links.	*See, e.g.,* Crouch, 1989, at 228-230
16. The method of claim 15 wherein the step of deriving comprises the step of choosing the top rated candidate cluster links.	*See, e.g.,* Crouch, 1989, at p. 230 Lists the value of the correlation measure of the query vector with either the centroid vector or the document vector associated with each node in the subtree. During the search process the user may change the correlation measure being calculated by means of the Correlation Measure pop-up menu. At present, the system provides a choice of several measures including vector product, inner product, Tanimoto, cosine and overlap. • Provides a listing of the concepts contained within the query vector (see also Fig. 6). This information is also displayed in the query window; however, in the tree display, the concepts in the query are displayed in ascending order of document frequency. The user may alter the query by adding or deleting concepts from the query vector during the search process without returning to the query window ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
18. A method of analyzing a database having objects and a first numerical representation of direct relationships in the database, comprising the steps of:	*See, e.g.,* Crouch, 1989, at pp. 226, 228-230 In hypertext information retrieval, each node is generally assumed to be a single document. Links exist which connect each document to other documents having keywords in common with it; the semantics of the links between nodes are keywords (document index terms) or some descriptive information representing the connected documents. In this paper we introduce an hierarchical structure which provides additional semantic information within and between nodes. This structure seems particularly well suited to the user's exploration of a document collection in a visual context. The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226)

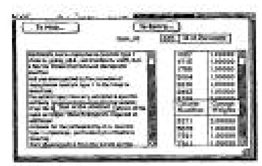
Claim Text from ’494 Patent	Crouch, 1989
	The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up the tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively in automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
generating a second numerical representation using the first numerical representation, wherein the second numerical representation accounts for indirect relationships in the database;	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are

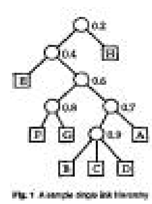
Claim Text from ’494 Patent	Crouch, 1989
	highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)  Figure 1 Fig. 1 A sample single link hierarchy (Crouch, 1989, p 229)

Claim Text from '494 Patent	Crouch, 1989
	A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
storing the second numerical representation;	*See, e.g.,* Crouch, 1989, at pp. 228- 230 The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. (Crouch, 1989, p. 228) Cluster hierarchies have been used effectively in automatic searches. Such hierarchies are

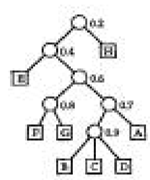
Claim Text from '494 Patent	Crouch, 1989
	also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
identifying at least one object in the database, wherein the stored numerical representation is used to identify objects; and	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of

Claim Text from ’494 Patent	Crouch, 1989
	similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node

Claim Text from ’494 Patent	Crouch, 1989
	being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229
displaying one or more identified objects from the database.	*See, e.g.,* Crouch, 1989, at p. 233, 234 Clicking on a terminal node (a document icon) results in the display of additional information associated with the document. (Crouch, 1989, p. 233) Figure 8  (Crouch, 1989, p. 234)
19. The method of claim 18 wherein the step of generating a second numerical representation comprises: selecting an object in the database for analysis;	*See, e.g.,* Crouch, 1989, at p. 228 The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. (Crouch, 1989, p. 228)
analyzing the direct relationships expressed by the first numerical representation for indirect relationships involving the selected object; and creating a second numerical representation of the direct and indirect relationships involving the selected object.	*See, e.g.,* Crouch, 1989, at pp. 228, 229, 230 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are

Claim Text from ’494 Patent	Crouch, 1989
	highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229)

Claim Text from ’494 Patent	Crouch, 1989
	A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
20. The method of 18 wherein the step of identifying at least one object in the database comprises: searching for objects in a database using the stored numerical representation, wherein direct and/or indirect relationships are searched.	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject

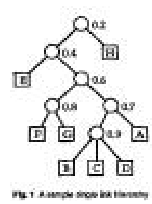
Claim Text from '494 Patent	Crouch, 1989
	descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search

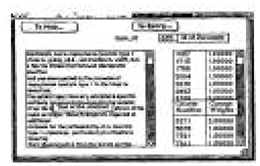
Claim Text from '494 Patent	Crouch, 1989
	will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
21. The method of claim 18 wherein the displaying step comprises: generating a graphical display for representing an object in the database.	*See, e.g.,* Crouch, 1989, at pp.226, 230 The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226) In general, a tree representation of a clustered collection is too large to be displayed in its entirely. Therefore, a user is presented with two views of the cluster tree simultaneously: a local view containing the subtree within which the user is currently browsing (see Fig. 4) and a global view, a more comprehensive view of the tree containing a significantly larger number of nodes than the local view (see Fig. 5). A user-directed traversal among the nodes is simultaneously reflected in both displays. The global view permits the user to observe where the search is being conducted in relation to the entire tree while the loca1 view provides the user with more detailed information about a specific subtree. (Crouch, 1989, p. 230) Provides a listing of the concepts contained within the query vector (see also Fig. 6). This information is also displayed in the query window; however, in the tree display, the concepts

Claim Text from '494 Patent	Crouch, 1989
	in the query are displayed in ascending order of document frequency. The user may alter the query by adding or deleting concepts from the query vector during the search process without returning to the query window. (Crouch, 1989, p. 230)
23. A method of representing data in a computer database with relationships, comprising the steps of:	*See, e.g.,* Crouch, 1989, at pp. 226, 228, 229 In hypertext information retrieval, each node is generally assumed to be a single document. Links exist which connect each document to other documents having keywords in common with it; the semantics of the links between nodes are keywords (document index terms) or some descriptive information representing the connected documents. In this paper we introduce an hierarchical structure which provides additional semantic information within and between nodes. This structure seems particularly well suited to the user's exploration of a document collection in a visual context. The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226) The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up the tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively in automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on

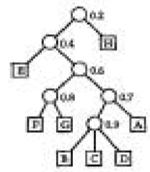
Claim Text from ’494 Patent	Crouch, 1989
	the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
assigning nodes node identifications;	*See, e.g.,* Crouch, 1989, at p. 230, Fig. 8 Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists;	*See, e.g.,* Crouch, 1989, at pp. 226, 228, 229 In hypertext information retrieval, each node is generally assumed to be a single document. Links exist which connect each document to other documents having keywords in common with it; the semantics of the links between nodes are keywords (document index terms) or some descriptive information representing the connected documents. In this paper we introduce an hierarchical structure which provides additional semantic information within and between nodes. This structure seems particularly well suited to the user’s exploration of a document collection in a visual context. The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226) The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up the tree until the similarity between the query and the

Claim Text from '494 Patent	Crouch, 1989
	parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively in automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
allocating a weight to each link, wherein the weight signifies the strength of the relationship represented by the link relative to the strength of other relationships represented by other links; and	*See, e.g.,* Crouch, 1989, at pp. 228, 229 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228)

Claim Text from ’494 Patent	Crouch, 1989
	To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228) Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations,

Claim Text from '494 Patent	Crouch, 1989
	we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
displaying a node identification.	*See, e.g.,* Crouch, 1989, at p. 234 In general, a tree representation of a clustered collection is too large to be displayed in its entirely. Therefore, a user is presented with two views of the cluster tree simultaneously: a local view containing the subtree within which the user is currently browsing (see Fig. 4) and a global view, a more comprehensive view of the tree containing a significantly larger number of nodes than the local view (see Fig. 5). A user-directed traversal among the nodes is simultaneously reflected in both displays. The global view permits the user to observe where the search is being conducted in relation to the entire tree while the loca1 view provides the user with more detailed information about a specific subtree. (Crouch, 1989, p. 230) Figure 8  (Crouch, 1989, p. 234)
24. The method of claim 23, wherein the data in the database is objects, wherein the nodes represent objects and each object is assigned a node identification, and wherein the relationships that exist comprise direct relationships between objects, further comprising the step of:	*See, e.g.,* Crouch, 1989, at pp. 228, 229, 230, Fig. 8 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in

Claim Text from ’494 Patent	Crouch, 1989
searching generated links, wherein nodes are located by searching the generated links.	which terminal nodes correspond to single documents and interior nodes to groups of documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)

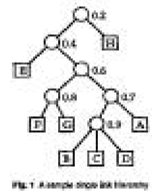
Claim Text from ’494 Patent	Crouch, 1989
	 Figure 1 Fig. 1 A sample single link hierarchy (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229) Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
25. The method of claim 23 further comprising the step of: generating link sub-types, comprising the steps of: identifying each link sub-type with a name; and providing a comment to one or more link subtypes.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants’ P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
31. The method of claim 23 wherein attributes are	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the

Claim Text from ’494 Patent	Crouch, 1989
assigned to nodes.	alleged invention, as evidenced by substantial other references identified in Defendants’ P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
32. The method of claim 31 further comprising the step of: generating node sub-types wherein the node sub-types are assigned information.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention, as evidenced by substantial other references identified in Defendants’ P. R. 3-3 statement and accompanying charts. Rather than repeat those disclosures here, they are incorporated by reference into this chart.
33. A method of representing data in a computer database and for computerized searching of the data, wherein relationships exist in the database, comprising:	*See, e.g.,* Crouch, 1989, at pp. 226, 228, 229 In hypertext information retrieval, each node is generally assumed to be a single document. Links exist which connect each document to other documents having keywords in common with it; the semantics of the links between nodes are keywords (document index terms) or some descriptive information representing the connected documents. In this paper we introduce an hierarchical structure which provides additional semantic information within and between nodes. This structure seems particularly well suited to the user’s exploration of a document collection in a visual context. The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226) The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up the tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high

Claim Text from ’494 Patent	Crouch, 1989
	kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively in automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
assigning links to represent relationships in the database;	*See, e.g.,* Crouch, 1989, at p. 226 In hypertext information retrieval, each node is generally assumed to be a single document. Links exist which connect each document to other documents having keywords in common with it; the semantics of the links between nodes are keywords (document index terms) or some descriptive information representing the connected documents. In this paper we introduce an hierarchical structure which provides additional semantic information within and between nodes. This structure seems particularly well suited to the user’s exploration of a document collection in a visual context. The user may browse among the data items by analyzing a graphical display of the structure itself as well as the semantic links between nodes. (Crouch, 1989, p. 226)
generating node identifications based upon the assigned links, wherein node identifications are generated so that each link represents a relationship between two identified nodes;	*See, e.g.,* Crouch, 1989, at pp. 228, 230 The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as

Claim Text from ’494 Patent	Crouch, 1989
	they fail certain criteria. (Crouch, 1989, p. 228) For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. (Crouch, 1989, p. 228) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
storing the links and node identifications, wherein the links and nodes may be retrieved;	*See, e.g.,* Crouch, 1989, at pp. 228, 230 The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. (Crouch, 1989, p. 228) Lists the number of documents contained within the subtree defined by eac hnode as well as the number of children of that node. ... Lists document identifiers represented by the leaf nodes of the tree. (Crouch, 1989, p. 230)
searching for node identifications using the stored links; and	*See, e.g.,* Crouch, 1989, at pp. 228- 230 Clustered Document Environments A principal advantage of the vector space model for use n hypertext information retrieval is that algorithms exist for structuring a document collection in such a manner that similar documents are grouped together. A cluster hierarchy is represented by a tree structure in which terminal nodes correspond to single documents and interior nodes to groups of

Claim Text from '494 Patent	Crouch, 1989
	documents. In a hypertext system based on a clustered environment, the user can readily focus his/her search on those groups (clusters) that are likely to contain documents which are highly similar to the query. Additionally, the cluster hierarchy is beneficial as a browsing tool in that it makes it possible easily to locate neighboring items with related subject descriptions. (Crouch, 1989, p. 228) Fig. 1 contains an example of a hierarchy for the single link agglomerative clustering method. In the single link method the similarity between two clusters is the maximum of the similarities between all pairs of documents such that one document of the pair is in one cluster and the other document is in the other cluster. It may be noted that in the hierarchy documents may appear at any level and that clusters overlap only in the sense that smaller clusters are nested within larger clusters. (Crouch, 1989, p. 228) To retrieve documents automatically in a clustered environment, comparisons are generally made between the query vector and document vectors using one of the standard measures of similarity. A cluster search simplifies the search process by limiting the search to subsets of documents. For example, with an agglomeratively clustered tree such as that shown in Fig. 1, a straightforward, narrow, depth-first search starts at the top of the tree and calculates the similarity between the query and each of its children. The child most similar to the query is selected, and the similarity between the query and each of the non-document children of that node is calculated. The process is repeated until either all the similarities between the query and the non-document children of some node are less than that between the query and the node itself, or all the children of that node are document nodes. The documents comprising the cluster represented by that node are returned. The search may be broadened by considering more than one path at each level. The broadest search considers all paths and abandons them as they fail certain criteria. (Crouch, 1989, p. 228) Each cluster in Fig. 1 is labelled with the level of association between the items under it. The clustering level determines the association strength of the corresponding items. Thus the similarity between items B, C andD in Fig. 4 is 0.9. On the other hand, the similarity between item A and the cluster containing items B, C and D is only 0.7. The level of association is a useful link semantic in a hypertext system. (Crouch, 1989, p. 228)

Claim Text from ’494 Patent	Crouch, 1989
	Figure 1  (Crouch, 1989, p 229) A bottom-up search may also be performed on such a tree. The cluster at the lowest level of the tree whose centroid is most similar to the query is chosen as the node at which the search will start. The search continues up he tree until the similarity between the query and the parent of the current node is smaller than the similarity between the query and the current node. The documents contained in the cluster corresponding to the current node are returned. The bottom-up searches are often more effective due to the uncertainty involved at high kevels of the hierarchy, [Crof80] Cluster hierarchies have been used effectively to automatic searches. Such hierarchies are also useful in performing searches based on browsing operations. These types of operations, we believe, can produce significant improvement in retrieval performance. Automatic cluster searches are highly structured; the next link in the search path is determined solely on the basis of the similarity between the query vector and the vector representation of the node being evaluated. By displaying suitable portions of the hierarchy during the course of the search operations and letting the user choose appropriate search paths at each point, the output obtained should be superior to that obtained by automatic cluster searching. (Crouch, 1989, p. 229)
displaying node identifications, wherein the displayed node identifications are located in the searching step.	*See, e.g.,* Crouch, 1989, at p. 234 In general, a tree representation of a clustered collection is too large to be displayed in its entirely. Therefore, a user is presented with two views of the cluster tree simultaneously: a local view containing the subtree within which the user is currently browsing (see Fig. 4) and a global view, a more comprehensive view of the tree containing a significantly larger number of nodes than the local view (see Fig. 5). A user-directed traversal among the nodes is simultaneously reflected in both displays. The global view permits the user to observe

Claim Text from '494 Patent	Crouch, 1989
	where the search is being conducted in relation to the entire tree while the loca1 view provides the user with more detailed information about a specific subtree. (Crouch, 1989, p. 230)  Figure 8 (Crouch, 1989, p. 234)

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR US PATENT NO. 5,832,494

BASED ON BOTAFOGO, R.A. "CLUSTER ANLAYSIS FOR HYPERTEXT SYSTEMS" ACM SIGIR '93, VOL. 6, 116-125 (1993). ("BOTAFOGO, 1993")

Claim Text from '494 Patent	Botafogo, 1993
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[1a] selecting a node for analysis;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[1b] generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[1c] deriving actual cluster links from the candidate cluster links;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[1d] identifying one or more nodes for display; and	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[1e] displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.,* Botafogo, 1993, at p. 121-122
2. The method of claim 1 wherein each link is given a length, the step of generating the candidate cluster links comprises the steps of:	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[2a] choosing a number as the maximum number of link lengths that will be examined; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[2b] examining only those links which are less than the maximum number of link lengths.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of: selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122

Claim Text from ’494 Patent	Botafogo, 1993
5. The method of claim 1 wherein the step of generating the candidate cluster links comprises the step of: eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[12a] selecting an object to determine the proximity of other objects to the selected object;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[12b] generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect relationships in the database;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[12c] deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[12d] displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
13. The method of 12 wherein a set of direct links exists for the database, and wherein the step of generating a candidate cluster link set comprises: recursively analyzing portions of the set of direct links for indirect links.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122

Claim Text from ’494 Patent	Botafogo, 1993
14. A method for representing the relationship between nodes using stored direct links, paths, and candidate cluster links, comprising the steps of:	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[14a] initializing a set of candidate cluster links;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[14b] selecting the destination node of a path as the selected node to analyze;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[14c] retrieving the set of direct links from the selected node to any other node in the database;	*See, e.g.,* Botafogo, 1993, at p. 118-119, 121-122
[14d] determining the weight of the path using the retrieved direct links;	*See, e.g.,* Botafogo, 1993, at p. 117-118
[14e] repeating steps b through d for each path; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[14f] storing the determined weights as candidate cluster links.	*See, e.g.,* Botafogo, 1993, at p. 117-118
15. The method of claim 14 further comprising the step of deriving the actual cluster links wherein the actual cluster links are a subset of the candidate cluster links.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
16. The method of claim 15 wherein the step of deriving comprises the step of choosing the top rated candidate cluster links.	*See, e.g.,* Botafogo, 1993, at p. 118-119, 121-122
18. A method of analyzing a database having objects and a first numerical representation of direct relationships in the database, comprising the steps	*See, e.g.,* Botafogo, 1993, at p. 118-119, 121-122

Claim Text from ’494 Patent	Botafogo, 1993
of:	
[18a] generating a second numerical representation using the first numerical representation, wherein the second numerical representation accounts for indirect relationships in the database;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[18b] storing the second numerical representation;	*See, e.g.,* Botafogo, 1993, at 118-119
[18c] identifying at least one object in the database, wherein the stored numerical representation is used to identify objects; and	*See, e.g.,* Botafogo, 1993, at 118-119, 121-122
[18d] displaying one or more identified objects from the database.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
19. The method of claim 18 wherein the step of generating a second numerical representation comprises: selecting an object in the database for analysis;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[19a] analyzing the direct relationships expressed by the first numerical representation for indirect relationships involving the selected object; and creating a second numerical representation of the direct and indirect relationships involving the selected object.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
20. The method of 18 wherein the step of identifying at least one object in the database comprises: searching for objects in a database using the stored numerical representation, wherein direct and/or	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122

Claim Text from ’494 Patent	Botafogo, 1993
indirect relationships are searched.	
21. The method of claim 18 wherein the displaying step comprises: generating a graphical display for representing an object in the database.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
23. A method of representing data in a computer database with relationships, comprising the steps of:	*See below:*
[23a] assigning nodes node identifications;	*See, e.g.,* Botafogo, 1993, at p. 116, 117, 119-21
[23b] generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[23c] allocating a weight to each link, wherein the weight signifies the strength of the relationship represented by the link relative to the strength of other relationships represented by other links; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[23d] displaying a node identification.	*See, e.g.,* Botafogo, 1993, at p. 121-122
24. The method of claim 23, wherein the data in the database is objects, wherein the nodes represent objects and each object is assigned a node identification, and wherein the relationships that exist comprise direct relationships between objects, further comprising the step of: searching generated links, wherein nodes are located by searching the generated links.	*See, e.g.,* Botafogo, 1993, at p. 116, 117-118, 119-22
25. The method of claim 23 further comprising the	*See below:*

Claim Text from ’494 Patent	Botafogo, 1993
step of: generating link sub-types, comprising the steps of:	
[25a] identifying each link sub-type with a name; and	*See, e.g.,* Botafogo, 1993, at p. 117
[25b] providing a comment to one or more link subtypes.	*See, e.g.,* Botafogo, 1993, at p. 117
31. The method of claim 23 wherein attributes are assigned to nodes.	*See, e.g.,* Botafogo, 1993, at p. 119
32. The method of claim 31 further comprising the step of: generating node sub-types wherein the node sub-types are assigned information.	*See, e.g.,* Botafogo, 1993, at p. 119
33. A method of representing data in a computer database and for computerized searching of the data, wherein relationships exist in the database, comprising:	*See below:*
[33a] assigning links to represent relationships in the database;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[33b] generating node identifications based upon the assigned links, wherein node identifications are generated so that each link represents a relationship between two identified nodes;	*See, e.g.,* Botafogo, 1993, at
[33c] storing the links and node identifications, wherein the links and nodes may be retrieved;	*See, e.g.,* Botafogo, 1993, at
[33d] searching for node identifications using the stored links; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[33e] displaying node identifications, wherein the	*See, e.g.,* Botafogo, 1993, at p. 121-122

Claim Text from ’494 Patent	Botafogo, 1993
displayed node identifications are located in the searching step.	

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,832,494 BASED ON "*NCSA MOSAIC AND THE WORLD WIDE WEB: GLOBAL HYPERMEDIA PROTOCOLS FOR THE INTERNET*," BRUCE SCHATZ & JOSEPH HARDIN (1994) ("SCHATZ, 1994")

Claim Text from '494 Patent	Shatz, 1974
7. The method of claim 1, wherein one or more nodes provide external connections to objects external to the database, the method further comprising the steps of:	*See* Schatz, e.g., at p. 895, 896
Activating the desired node; and	*See* Schatz, e.g., at p. 895, 896, and 897
Accessing the external object linked to the node.	*See* Schatz, e.g., at 895, 896
8. The method of claim 7, wherein the external object is an independent application which can be executed in background, the method further comprising the step of:	*See* Schatz, e.g., at p. 895, 896, and 898
executing the independent application.	*See* Schatz, e.g., at p. 896, 897, and 898
9. The method of claim 8, wherein one or more nodes provide links to more than one independent application which can be executed as an extension, the method further comprising the steps of:	*See* Schatz, e.g., at p. 896
displaying a list of independent applications linked to the node, wherein the step of accessing accesses an independent application.	*See* Schatz, e.g., at p. 896
10. The method of claim 8, wherein the connection provides the independent application access to the information stored within the database.	*See* Schatz, e.g., at p. 896, 897
11. The method of claim 7, wherein the external connection is to another computer, wherein	*See* Schatz, e.g., at p. 895, 896

Claim Text from ’494 Patent	Shatz, 1974
information is located that can be accessed, the step of accessing further comprising the step of:	
accessing the information located within the computer.	*See* Schatz, e.g., at p. 895, 896

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims as appropriate, for example, depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

Invalidity Claim Chart for US Patent No. 5,832,494

Based on Gerard Salton and Chris Buckley, "Automatic Text Structuring and Retrieval – Experiments in Automatic Encyclopedia Searching ("Salton & Buckley 1991")

Claim Text from '494 Patent	Salton & Buckley 1991
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.*, Salton & Buckley 1991 at 22 ("An identification of semantically homogenous text excerpts leads to the generation of text links between related text portions. Such links transform linear texts into structured text representations that provide selective text reading and traversal paths by following the available content links."); *id.* ("Network structures are often used, in which case the concepts of interest in a subject area are represented by network nodes, and the main relationships between concepts by network branches").
[1a] selecting a node for analysis;	*See, e.g.*, Salton & Buckley 1991 at 23 ("Each available text (including query as well as document texts) is broken down into individual text units . . . A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment. These term vectors form the basis for the text comparison operations"); 25 ("A standard encyclopedia search for a one-paragraph query (document 114, Acacia) is illustrated in Table 1.").
[1b] generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.*, Salton & Buckley 1991 at 23 ("Similarities between particular text items (or between text items and information requests) are obtained by comparing the term vectors for pairs of text items at various levels of detail. When sufficient similarities are detected in both global as well as local contexts, the texts are assumed to be related.").
[1c] deriving actual cluster links from the candidate cluster links;	*See, e.g.*, Salton & Buckley 1991 at 23 ("In practice, two text sections might then be related when the similarity between the vectors describing the text sections exceeds a stated threshold, and in addition the sections also contain at least one paragraph pair with a

Claim Text from '494 Patent	SALTON & BUCKLEY 1991
	sufficiently large paragraph similarity.").
[1d] identifying one or more nodes for display; and	*See* Chart for Claim 1[e], *infra*.
[1e] displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.*, Salton & Buckley 1991 at Tables 1-5.
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of: selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	*See* Chart for Claim 1[c], *supra*.
5. The method of claim 1 wherein the step of generating the candidate cluster links comprises the step of: eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.	Inherently disclosed by Chart for Claim 1[c], *supra*.
12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:	*See, e.g.*, Salton & Buckley 1991 at 24 (disclosing a stored database of 24,900 objects); Chart for Claim 1, *supra* (disclosing links between said objects); Tables 1-5 (disclosing the display of said objects).
[12a] selecting an object to determine the proximity of other objects to the selected object;	*See, e.g.*, Salton & Buckley 1991 at 25 ("A standard encyclopedia search for a one-paragraph query (document 114, Acacia) is illustrated in Table 1.").

Claim Text from ’494 Patent	**SALTON & BUCKLEY 1991**
[12b] generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect relationships in the database;	*See, e.g.*, Salton & Buckley 1991 at 25 (disclosing first- and second-level searches starting from the initial “Acacia” object); Chart for Claim 1, *supra* (disclosing a link network between objects).
[12c] deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and	*See* Chart for Claim 1[c], *supra*.
[12d] displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See*, Salton & Buckley 1991 at Tables 1-5.
13. The method of 12 wherein a set of direct links exists for the database, and wherein the step of generating a candidate cluster link set comprises: recursively analyzing portions of the set of direct links for indirect links.	*See, e.g.*, Salton & Buckley 1991 at 25 (“A multi-stage search strategy is used where all articles with a global query similarity exceeding 0.20 are retrieved initially. The retrieved items (documents 15552, Mimosa and 11949, Indigo Plant) are then separately used as queries for second-stage searches with an increased retrieval threshold of 0.25.”).
14. A method for representing the relationship between nodes using stored direct links, paths, and candidate cluster links, comprising the steps of:	*See*, Chart for Claim 1, *supra*.
15. The method of claim 14 further comprising the step of deriving the actual cluster links wherein the actual cluster links are a subset of the candidate cluster links.	Inherently disclosed by Chart for Claim 1, *supra*.
16. The method of claim 15 wherein the step of deriving comprises the step of choosing the top rated candidate cluster links.	Inherently disclosed by Chart for Claim 1, *supra*.
[18d] displaying one or more identified objects from the database.	*See, e.g.*, Salton & Buckley 1991 at Tables 1-5.

Claim Text from '494 Patent	SALTON & BUCKLEY 1991
19. The method of claim 18 wherein the step of generating a second numerical representation comprises: selecting an object in the database for analysis;	*See, e.g.*, Salton & Buckley 1991 at 23 ("Each available text (including query as well as document texts) is broken down into individual text units . . . A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment. These term vectors form the basis for the text comparison operations"); 25 ("A standard encyclopedia search for a one-paragraph query (document 114, Acacia) is illustrated in Table 1.").
[19a] analyzing the direct relationships expressed by the first numerical representation for indirect relationships involving the selected object; and creating a second numerical representation of the direct and indirect relationships involving the selected object.	*See, e.g.*, Salton & Buckley at 25 (disclosing first- and second-level searches from an initial selected object).
20. The method of 18 wherein the step of identifying at least one object in the database comprises: searching for objects in a database using the stored numerical representation, wherein direct and/or indirect relationships are searched.	*See* Chart for Claim 19[a], *supra*.
21. The method of claim 18 wherein the displaying step comprises: generating a graphical display for representing an object in the database.	*See* Salton & Buckley 1991 at Table 1-5.
23. A method of representing data in a computer database with relationships, comprising the steps of:	*See* Salton & Buckley 1991 at 24 ("This database consists of about 24,900 articles of text . . .[a]n automated encyclopedia search system is implemented which uses particular encyclopedia articles as search requests, and retrieves related articles in decreasing order of presumed similarity with the request articles.").
[23a] assigning nodes node identifications;	*See, e.g.*, Salton & Buckley 1991 at 23 ("A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence, etc.) a set of weighted terms to be used for content identification of the corresponding text fragment.").

Claim Text from ’494 Patent	SALTON & BUCKLEY 1991
[23b] generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists;	*See, e.g.*, Salton & Buckley 1991 at 22 (“An identification of semantically homogenous text excerpts leads to the generation of text links between related text portions. Such links transform linear texts into structured text representations that provide selective text reading and traversal paths by following the available content links.”).
[23d] displaying a node identification.	*See, e.g.*, Salton & Buckley 1991 at Tables 1-5.
24. The method of claim 23, wherein the data in the database is objects, wherein the nodes represent objects and each object is assigned a node identification, and wherein the relationships that exist comprise direct relationships between objects, further comprising the step of: searching generated links, wherein nodes are located by searching the generated links.	*See, e.g.*, Salton & Buckley 1991 at 22 (“links transform linear texts into structured text representations that provide selective text reading and traversal paths by following the available content links. In addition, a recognition of semantically related text portions also leads to the retrieval of relevant texts in answer to available search requests, because close similarities between document and query texts may be indicative of a relevance relationship between them.”).
31. The method of claim 23 wherein attributes are assigned to nodes.	*See, e.g.*, Salton & Buckley 1991 at 23 (“A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence etc.) a set of weighted terms to be used for content identification of the corresponding text fragment.”).
32. The method of claim 31 further comprising the step of: generating node sub-types wherein the node sub-types are assigned information.	*See, e.g.*, Salton & Buckley 1991 at 23 (“Each available text . . . is broken down into individual text units – for example, text sections, text paragraphs, and individual sentences. A standard indexing system is used to assign to each text unit (that is, section, paragraph, sentence etc.) a set of weighted terms to be used for content identification of the corresponding text fragment.”).
33. A method of representing data in a computer database and for computerized searching of the data, wherein relationships exist in the database,	*See, e.g.*, Salton & Buckley 1991 at 24 (“An automated encyclopedia search system is implemented which uses particular encyclopedia articles as search requests, and retrieves related articles in decreasing order of presumed similarity with the request articles.”).

Claim Text from ’494 Patent	SALTON & BUCKLEY 1991
comprising:	
[33a] assigning links to represent relationships in the database;	*See* Chart for Claim 1, *supra*.
[33b] generating node identifications based upon the assigned links, wherein node identifications are generated so that each link represents a relationship between two identified nodes;	*See*, *e.g.*, Salton & Buckley 1991 at 22 (“Such links transform linear texts into structured text representations . . . .”).
[33c] storing the links and node identifications, wherein the links and nodes may be retrieved;	Inherently disclosed by the use of automated database and automated retrieval methods.
[33d] searching for node identifications using the stored links; and	*See*, *e.g.*, Salton & Buckley 1991 at 22 (“links transform linear texts into structured text representations that provide selective text reading and traversal paths by following the available content links. In addition, a recognition of semantically related text portions also leads to the retrieval of relevant texts in answer to available search requests”).
[33e] displaying node identifications, wherein the displayed node identifications are located in the searching step.	*See*, *e.g.*, Salton & Buckley 1991 at Tables 1-5.

Invalidity Claim Chart for US Patent No. 5,832,494
Based on Botafogo, R.A. et al. "Identifying Aggregates in Hypertext Structures" Hypertext '91 Proceedings, 63-74 (1991). ("Botafogo, 1991")

Claim Text from '494 Patent	Botafogo, 1991
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.,* Botafogo, 1991, at p. 64-66
[1a] selecting a node for analysis;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[1b] generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.,* Botafogo, 1991, at p. 65-66, 68, 70, 72
[1c] deriving actual cluster links from the candidate cluster links;	*See, e.g.,* Botafogo, 1991, at p. 65-66, 68, 70, 72
[1d] identifying one or more nodes for display; and	*See, e.g.,* Botafogo, 1991, at p. 71-72
[1e] displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.,* Botafogo, 1991, at p. 71-72
2. The method of claim 1 wherein each link is given a length, the step of generating the candidate cluster links comprises the steps of:	*See, e.g.,* Botafogo, 1991, at p. 64-66
[2a] choosing a number as the maximum number of link lengths that will be examined; and	*See, e.g.,* Botafogo, 1991, at p. 64-66
[2b] examining only those links which are less than the maximum number of link lengths.	*See, e.g.,* Botafogo, 1991, at p. 64-66
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of: selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	*See, e.g.,* Botafogo, 1991, at p. 66, 68, 70

Claim Text from ’494 Patent	Botafogo, 1991
5. The method of claim 1 wherein the step of generating the candidate cluster links comprises the step of: eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.	*See, e.g.,* Botafogo, 1991, at p. 66, 68, 70, 72
12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:	*See, e.g.,* Botafogo, 1991, at p. 64-66
[12a] selecting an object to determine the proximity of other objects to the selected object;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[12b] generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect relationships in the database;	*See, e.g.,* Botafogo, 1991, at p. 65-66, 68, 70, 72
[12c] deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and	*See, e.g.,* Botafogo, 1991, at p. 65-66, 68, 70, 72
[12d] displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See, e.g.,* Botafogo, 1991, at p. 71- 72
13. The method of 12 wherein a set of direct links exists for the database, and wherein the step of generating a candidate cluster link set comprises: recursively analyzing portions of the set of direct links for indirect links.	*See, e.g.,* Botafogo, 1991, at p. 65-66, 68, 70, 72

Claim Text from '494 Patent	Botafogo, 1991
14. A method for representing the relationship between nodes using stored direct links, paths, and candidate cluster links, comprising the steps of:	*See, e.g.,* Botafogo, 1991, at p. 64-66
[14a] initializing a set of candidate cluster links;	*See, e.g.,* Botafogo, 1991, at p. 66, 68, 70
[14b] selecting the destination node of a path as the selected node to analyze;	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[14c] retrieving the set of direct links from the selected node to any other node in the database;	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[14d] determining the weight of the path using the retrieved direct links;	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[14e] repeating steps b through d for each path; and	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[14f] storing the determined weights as candidate cluster links.	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
15. The method of claim 14 further comprising the step of deriving the actual cluster links wherein the actual cluster links are a subset of the candidate cluster links.	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
16. The method of claim 15 wherein the step of deriving comprises the step of choosing the top rated candidate cluster links.	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
18. A method of analyzing a database having objects and a first numerical representation of direct relationships in the database, comprising the steps	*See, e.g.,* Botafogo, 1991, at p. 64-66

Claim Text from ’494 Patent	Botafogo, 1991
of:	
[18a] generating a second numerical representation using the first numerical representation, wherein the second numerical representation accounts for indirect relationships in the database;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[18b] storing the second numerical representation;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[18c] identifying at least one object in the database, wherein the stored numerical representation is used to identify objects; and	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[18d] displaying one or more identified objects from the database.	*See, e.g.,* Botafogo, 1991, at p. 71-72
19. The method of claim 18 wherein the step of generating a second numerical representation comprises: selecting an object in the database for analysis;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[19a] analyzing the direct relationships expressed by the first numerical representation for indirect relationships involving the selected object; and creating a second numerical representation of the direct and indirect relationships involving the selected object.	*See, e.g.,* Botafogo, 1991, at p. 64-66
20. The method of 18 wherein the step of identifying at least one object in the database comprises: searching for objects in a database using the stored numerical representation, wherein direct and/or	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70

Claim Text from '494 Patent	Botafogo, 1991
indirect relationships are searched.	
21. The method of claim 18 wherein the displaying step comprises: generating a graphical display for representing an object in the database.	*See, e.g.,* Botafogo, 1991, at p. 70-72
23. A method of representing data in a computer database with relationships, comprising the steps of:	*See below:*
[23a] assigning nodes node identifications;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[23b] generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[23c] allocating a weight to each link, wherein the weight signifies the strength of the relationship represented by the link relative to the strength of other relationships represented by other links; and	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[23d] displaying a node identification.	*See, e.g.,* Botafogo, 1991, at p. 70-72
24. The method of claim 23, wherein the data in the database is objects, wherein the nodes represent objects and each object is assigned a node identification, and wherein the relationships that exist comprise direct relationships between objects, further comprising the step of: searching generated links, wherein nodes are located by searching the generated links.	*See, e.g.,* Botafogo, 1991, at p. 66, 68, 70, 72
25. The method of claim 23 further comprising the	*See below:*

Claim Text from ’494 Patent	Botafogo, 1991
step of: generating link sub-types, comprising the steps of:	
[25a] identifying each link sub-type with a name; and	*See, e.g.,* Botafogo, 1991, at p. 67, 71
[25b] providing a comment to one or more link subtypes.	*See, e.g.,* Botafogo, 1991, at p. 67, 71
31. The method of claim 23 wherein attributes are assigned to nodes.	*See, e.g.,* Botafogo, 1991, at p. 67, 71
32. The method of claim 31 further comprising the step of: generating node sub-types wherein the node sub-types are assigned information.	*See, e.g.,* Botafogo, 1991, at p. 67, 71
33. A method of representing data in a computer database and for computerized searching of the data, wherein relationships exist in the database, comprising:	*See below:*
[33a] assigning links to represent relationships in the database;	*See, e.g.,* Botafogo, 1991, at p. 64-66
[33b] generating node identifications based upon the assigned links, wherein node identifications are generated so that each link represents a relationship between two identified nodes;	*See, e.g.,* Botafogo, 1991, at 64-66
[33c] storing the links and node identifications, wherein the links and nodes may be retrieved;	*See, e.g.,* Botafogo, 1991, at 64-66
[33d] searching for node identifications using the stored links; and	*See, e.g.,* Botafogo, 1991, at p. 64-66, 68, 70
[33e] displaying node identifications, wherein the	*See, e.g.,* Botafogo, 1991, at p. 70-72

Claim Text from ’494 Patent	Botafogo, 1991
displayed node identifications are located in the searching step.	

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 5,832,494

Based on Joachims, T et al., "WebWatcher: Machine Learning and Hypertext" Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, 1995 ("Joachims 1995")

Claim Text from '494 Patent	Joachims 1995
1. A method of analyzing a database with indirect relationships, using links and nodes, comprising the steps of:	*See, e.g.,* p. 1, 3-5
[1a] Selecting a node for analysis;	*See, e.g.,* p.1 (e.g. "The target function we want to learn is a mapping from an arbitrary web page to a set of related pages.").
[1b] Generating candidate cluster links for the selected node, wherein the step of generating comprises an analysis of one or more indirect relationships in the database;	*See, e.g.,* p. 4-5 (e.g., "3.3 Algorithm")
[1c] Deriving actual cluster links from the candidate cluster links;	
[1d] identifying one or more nodes for display; and	*See, e.g.,* p. 4 (e.g., "The pages associated with the n most similar columns are returned by Related.").
[1e] displaying the identity of one or more nodes using the actual cluster links.	*See, e.g.,* p. 4
2. The method of claim 1 wherein each link is given a length, the step of generating the candidate cluster links comprises the steps of:	*See, e.g.,* p. 4
[2a] Choosing a number as the maximum number of link lengths that will be examined; and	*See, e.g.,* p. 4
[2b] examining only those links which are less than the maximum number of link lengths.	
3. The method of claim 1 wherein the step of deriving actual cluster links comprises the step of:	*See, e.g.,* p. 4 (e.g., "The pages associated with the n most similar columns are returned by Related.").

Claim Text from '494 Patent	Joachims 1995
selecting the top rated candidate cluster links, wherein the top rated candidate cluster links are those which are most closely linked to the node under analysis.	
5. The method of claim 1 wherein the step of generating the candidate cluster links comprises the step of: eliminating candidate cluster links, wherein the number of candidate cluster links is limited and the closest candidate cluster links are chosen over the remaining links.	*See, e.g.,* p. 4-5 (e.g., "The pages associated with the n most similar columns are returned by Related.").
7. The method of claim 1, wherein one or more nodes provide external connections to objects external to the database, the method further comprising the steps of:	*See, e.g.,* p. 1-3
[7a] Activating the desired node; and	
[7b] Accessing the external object linked to the node.	
11. The method of claim 7, wherein the external connection is to another computer, wherein information is located that can be accessed, the step of accessing further comprising the step of:	*See, e.g.,* p. 1-3
[11a] accessing the information located within the computer.	
12. A method for determining the proximity of an object in a stored database to another object in the stored database using indirect relationships, links, and a display, comprising:	*See, e.g.,* p. 1-4
[12a] Selecting an object to determine the proximity of other objects to the selected object;	*See, e.g.,* p.4 (e.g. "The target function we want to learn is a mapping from an arbitrary web page to a set of related pages.").

Claim Text from '494 Patent	Joachims 1995
[12b] generating a candidate cluster link set for the selected object, wherein the generating step includes an analysis of one or more indirect relationships in the database;	*See, e.g.,* p. 4-5 (e.g., "3.3 Algorithm")
[12c] Deriving an actual cluster link set for the selected object using the generated candidate cluster link set; and	
[12d] Displaying one or more of the objects in the database, referred to in the actual cluster link set, on a display.	*See, e.g.,* p. 4 (e.g., "The pages associated with the n most similar columns are returned by Related.").
23. A method of representing data in a computer database with relationships, comprising the steps of:	*See, e.g.,* p. 1-3
[23a] assigning nodes node identifications;	*See, e.g.,* p. 4-5 (e.g., "3.3 Algorithm")
[23b] generating links, wherein each link represents a relationship between two nodes and is identified by the two nodes in which the relationship exists;	*See, e.g.,* p. 4-5 (e.g., "3.3 Algorithm")
[23c] allocating a weight to each link, wherein the weight signifies the strength of the relationship represented by the link relative to the strength of other relationships represented by other links; and	*See, e.g.,* p. 4-5 (e.g., "3.3 Algorithm")
[23d] displaying a node identification.	*See, e.g.,* p. 4 (e.g., "The pages associated with the n most similar columns are returned by Related.").
24. The method of claim 23, wherein the data in the database is objects, wherein the nodes represent objects and each object is assigned a node identification, and wherein the relationships that exist comprise direct relationships between objects, further comprising the step of: searching generated links, wherein nodes are located by searching the generated links.	*See, e.g.,* p. 4-5 (e.g., "3.3 Algorithm")
25. The method of claim 23 further comprising the	*See, e.g.,* p. 3

Claim Text from ’494 Patent	Joachims 1995
step of: generating link sub-types, comprising the steps of:	
[25a] identifying each link sub-type with a name; and	*See, e.g.*, p. 3
[25b] Providing a comment to one or more link subtypes.	*See, e.g.*, p. 3
31. The method of claim 23 wherein attributes are assigned to nodes.	*See, e.g.*, p. 3
32. The method of claim 31 further comprising the step of: generating node sub-types wherein the node sub-types are assigned information.	*See, e.g.*, p. 3
33. A method of representing data in a computer database and for computerized searching of the data, wherein relationships exist in the database, comprising:	*See, e.g.*, p. 1-3
[33a] assigning links to represent relationships in the database;	*See, e.g.*, p. 3
[33b] generating node identifications based upon the assigned links, wherein node identifications are generated so that each link represents a relationship between two identified nodes;	*See, e.g.*, p. 4-5 (e.g., “3.3 Algorithm”)
[33c] storing the links and node identifications, wherein the links and nodes may be retrieved;	*See, e.g.*, p. 4-5 (e.g., “3.3 Algorithm”)
[33d] searching for node identifications using the stored links; and	*See, e.g.*, p. 4-5 (e.g., “3.3 Algorithm”)
[33e] displaying node identifications, wherein the displayed node identifications are located in the searching step.	*See, e.g.*, p. 2-3

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 6,233,571

BASED ON BENNY BRODDA, HANS KARLGREN, "CITATION INDEX AND MEASURES OF ASSOCIATION IN MECHANIZED DOCUMENT RETRIEVAL," KVAL PM 295 (1967). REPORT NO. 2 TO THE ROYAL TREASURY. PUBLISHED BY SPRAKFORLAGET SKRIPTOR. ("BRODDA & KARLGREN, 1967")

Claim Text from '571 Patent	Brodda & Karlgren, 1967
1. A method for using active links within the data of an object stored in a database of a computer so that a user may jump from viewing the data of the object in the database to a position outside the object in the database and outside the computer, comprising:	*See, e.g.*, Brodda & Karlgren, 1967, at pp. 1, 3
[1a] storing one or more links within data of the object in the database to positions outside of the computer, wherein the stored links are active links;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[1b] displaying the data of the object within the database, wherein one or more active links are displayed with the data from the object in the database, wherein positions are nodes in a network that may be accessed, the active links including hyperjump links between nodes in the network and the objects, and the step of displaying comprises:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6
[1c] generating a source map, wherein the source map represents hyperjump links that identify a chosen node as a destination of a link, and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 5
[1d] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link;	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[1e] selecting one of the displayed active links from those displayed with the displayed data; and	
[1f] jumping to the position outside the object in the	

Claim Text from ’571 Patent	Brodda & Karlgren, 1967
database.	
5. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6
[5a] choosing a node	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[5b] accessing the hyperjump data; ǀ Identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[5c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises proximity analyzing the identified hyperjump data; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4, 5, 8
[5d] displaying one or more determined hyperjump data.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
6. The method of claim 5, wherein the hyperjump data includes pointers and wherein the direct reference is a pointer pointing to the chosen node or from the chosen node, and the step of determining comprises analyzing the pointers.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2-4, 6
7. The method of claim 5, wherein the node represents a topic, the determined hyperjump data has a relationship to the topic, and the step of displaying displays determined hyperjump data that has a relationship to the topic.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
8. The method of claim 5, wherein the node is a web page in the network, the accessed hyperjump data are Universal Resource Locators of linked pages, and the step of determining hyperjump data	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 3-4

Claim Text from '571 Patent	Brodda & Karlgren, 1967
comprises analyzing the identified hyperjump data.	
9. The method of claim 5, wherein the node is a document in the network and the determined hyperjump data has a relationship to the document, the step of displaying comprising the step of listing the hyperjump data that has a relationship to the document.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6
11. The method of claim 5, wherein the nodes are nodes in the network that may be accessed, the hyperjump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
[11a] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 5
[11b] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
12. A method for visually displaying data related to a web having identifiable web pages and Universal Resource Locators with pointers, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 3-4, 6
[12a] choosing an identifiable web page;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[12b] identifying Universal Resource Locators for the web pages, wherein the identified Universal Resource Locators either point to or point away from the chosen web page;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 3
[12c] analyzing Universal Resource Locators,	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8

Claim Text from ’571 Patent	Brodda & Karlgren, 1967
including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen web page are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	
[12d] displaying identities of web pages, wherein the located Universal Resource Locators are used to identify web pages.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2-4, 6, 10
15. The method of claim 12, wherein the step of displaying the identities of web pages comprises generating a graphical user display wherein information within the Universal Resource Locators is parsed and used to generate the graphical user display.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 6
16. A method for navigating documents on the World Wide Web, comprising: | choosing a document;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-4
[16a] identifying documents that have a direct relationship to the chosen document;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6
[16b] locating documents that have an indirect relationship to the chosen document identifying Universal Resource Locators for the documents, wherein the identified Universal Resource Locators either point to or point away from the chosen document; analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen document are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 8

Claim Text from '571 Patent	Brodda & Karlgren, 1967
[16c] displaying a located document.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
17. The method of claim 16, wherein pages and their respective Universal Resource Locators are used and the step of locating documents comprises analyzing the pages and their respective Universal Resource Locators.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8
18. The method of claim 17, wherein the step of analyzing pages comprises cluster analyzing the pages.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 8
19. The method of claim 16, wherein the step of displaying a located document comprises:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
[19a] generating a screen display of identities of one or more located documents; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6
[19b] selecting one or more of the located documents.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
20. The method of claim 19, wherein the step of generating a screen display comprises generating a graphical display.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 4, 6, 10
21. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
[21a] choosing a node;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[21b] accessing the hyperjump data; | identifying hyperjump data from within the accessed	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 3, 4, 6

Claim Text from ’571 Patent	Brodda & Karlgren, 1967
hyperjump data that has a direct reference to the chosen node;	
[21c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises cluster analyzing the hyperjump data; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2, 5, 8
[21d] displaying one or more determined hyperjump data.	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
22. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
[22a] choosing a node;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1, 2
[22b] accessing the hyperjump data; | identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 3
[22c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data; and	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 1-5, 8
[22d] displaying one or more determined hyperjump data, wherein the nodes are nodes in the network that may be accessed, the hypejump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2, 4, 6, 10
[22e] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link, and wherein	*See, e.g.,* Brodda & Karlgren, 1967, at pp. 2-4, 6 Further, disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the

Claim Text from ’571 Patent	Brodda & Karlgren, 1967
the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	time of the alleged invention.

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

Invalidity Claim Chart for U.S. Patent No. 6,233,571
Based on Botafogo, R.A. "Cluster Anlaysis for Hypertext Systems" ACM SIGIR '93, Vol. 6, 116-125 (1993). ("Botafogo, 1993")

Claim Text from '571 Patent	Botafogo, 1993
1. A method for using active links within the data of an object stored in a database of a computer so that a user may jump from viewing the data of the object in the database to a position outside the object in the database and outside the computer, comprising:	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
[1a] storing one or more links within data of the object in the database to positions outside of the computer, wherein the stored links are active links;	*See, e.g.,* Botafogo, 1993, at 119-121
[1b] displaying the data of the object within the database, wherein one or more active links are displayed with the data from the object in the database, wherein positions are nodes in a network that may be accessed, the active links including hyperjump links between nodes in the network and the objects, and the step of displaying comprises:	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[1c] generating a source map, wherein the source map represents hyperjump links that identify a chosen node as a destination of a link, and	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[1d] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link;	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
[1e] selecting one of the displayed active links from those displayed with the displayed data; and	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122

Claim Text from '571 Patent	Botafogo, 1993
[1f] jumping to the position outside the object in the database.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
3. The method of claim 1, wherein the active links are embedded text and wherein the step of selecting comprises activating the embedded text.	*See, e.g.,* Botafogo, 1993, at 119-120
4. The method of claim 1, wherein computer software is used, further comprising: | generating an active link, wherein the active link can be used to jump from a location in the database to another database.	*See, e.g.,* Botafogo, 1993, at 119-120
5. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Botafogo, 1993, at p. 121-122
[5a] choosing a node	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[5b] accessing the hyperjump data; | Identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Botafogo, 1993, at 117-119
[5c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises proximity analyzing the identified hyperjump data; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[5d] displaying one or more determined hyperjump data.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122

Claim Text from ’571 Patent	Botafogo, 1993
6. The method of claim 5, wherein the hyperjump data includes pointers and wherein the direct reference is a pointer pointing to the chosen node or from the chosen node, and the step of determining comprises analyzing the pointers.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
7. The method of claim 5, wherein the node represents a topic, the determined hyperjump data has a relationship to the topic, and the step of displaying displays determined hyperjump data that has a relationship to the topic.	*See, e.g.,* Botafogo, 1993, at p. 119
8. The method of claim 5, wherein the node is a web page in the network, the accessed hyperjump data are Universal Resource Locators of linked pages, and the step of determining hyperjump data comprises analyzing the identified hyperjump data.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
9. The method of claim 5, wherein the node is a document in the network and the determined hyperjump data has a relationship to the document, the step of displaying comprising the step of listing the hyperjump data that has a relationship to the document.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
10. The method of claim 5, wherein the step of displaying comprises generating a graphical user display, and wherein information is displayed on a graphical display visually representing more than one coordinate plane.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
11. The method of claim 5, wherein the nodes are nodes in the network that may be accessed, the hyperjump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See below*

Claim Text from ’571 Patent	Botafogo, 1993
[11a] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link; and	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[11b] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
12. A method for visually displaying data related to a web having identifiable web pages and Universal Resource Locators with pointers, comprising:	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
[12a] choosing an identifiable web page;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[12b] identifying Universal Resource Locators for the web pages, wherein the identified Universal Resource Locators either point to or point away from the chosen web page;	*See, e.g.,* Botafogo, 1993, at 117-119
[12c] analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen web page are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[12d] displaying identities of web pages, wherein the located Universal Resource Locators are used to identify web pages.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
13. The method of claim 12, further comprising selecting a web page using the displayed identities	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122

Claim Text from ’571 Patent	Botafogo, 1993
of web pages.	
14. The method of claim 12, further comprising hyperjumping to the selected web page.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
15. The method of claim 12, wherein the step of displaying the identities of web pages comprises generating a graphical user display wherein information within the Universal Resource Locators is parsed and used to generate the graphical user display.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
16. A method for navigating documents on the World Wide Web, comprising: | choosing a document;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[16a] identifying documents that have a direct relationship to the chosen document;	*See, e.g.,* Botafogo, 1993, at 117-119
[16b] locating documents that have an indirect relationship to the chosen document identifying Universal Resource Locators for the documents, wherein the identified Universal Resource Locators either point to or point away from the chosen document; analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen document are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[16c] displaying a located document.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122

Claim Text from ’571 Patent	Botafogo, 1993
17. The method of claim 16, wherein pages and their respective Universal Resource Locators are used and the step of locating documents comprises analyzing the pages and their respective Universal Resource Locators.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
18. The method of claim 17, wherein the step of analyzing pages comprises cluster analyzing the pages.	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
19. The method of claim 16, wherein the step of displaying a located document comprises:	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[19a] generating a screen display of identities of one or more located documents; and	*See, e.g.,* Botafogo, 1993, at p. 121-122
[19b] selecting one or more of the located documents.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122
20. The method of claim 19, wherein the step of generating a screen display comprises generating a graphical display.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
21. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[21a] choosing a node;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[21b] accessing the hyperjump data; | identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the	*See, e.g.,* Botafogo, 1993, at 117-119

Claim Text from ’571 Patent	Botafogo, 1993
chosen node;	
[21c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises cluster analyzing the hyperjump data; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[21d] displaying one or more determined hyperjump data.	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
22. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122
[22a] choosing a node;	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[22b] accessing the hyperjump data; ǀ identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Botafogo, 1993, at 117-119
[22c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data; and	*See, e.g.,* Botafogo, 1993, at p. 117-118, 121-122
[22d] displaying one or more determined hyperjump data, wherein the nodes are nodes in the network that may be accessed, the hypejump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See, e.g.,* Botafogo, 1993, at p. 117, 121-122

Claim Text from ’571 Patent	Botafogo, 1993
[22e] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link, and wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	*See, e.g.,* Botafogo, 1993, at p. 117-119, 121-122

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 6,233,571

BASED ON CROUCH, D., CROUCH, C., ANDREAS, G., "THE USE OF CLUSTER HIERARCHIES IN HYPERTEXT INFORMATION RETRIEVAL," IN HYPERTEXT '89 PROCEEDINGS, SIGCHI BULLETIN, PP. 225-237, NOVEMBER 1989. ("CROUCH, 1989")

Claim Text from '571 Patent	Crouch, 1989
1. A method for using active links within the data of an object stored in a database of a computer so that a user may jump from viewing the data of the object in the database to a position outside the object in the database and outside the computer, comprising:	*See, e.g.,* Crouch, 1989, at pp. 226, 233
[1a] storing one or more links within data of the object in the database to positions outside of the computer, wherein the stored links are active links;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[1b] displaying the data of the object within the database, wherein one or more active links are displayed with the data from the object in the database, wherein positions are nodes in a network that may be accessed, the active links including hyperjump links between nodes in the network and the objects, and the step of displaying comprises:	*See, e.g.,* Crouch, 1989, at pp. 226, 233, 234
[1c] generating a source map, wherein the source map represents hyperjump links that identify a chosen node as a destination of a link, and	*See, e.g.,* Crouch, 1989, at pp.226, 230
[1d] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link;	*See, e.g.,* Crouch, 1989, at pp. 226, 233
[1e] selecting one of the displayed active links from those displayed with the displayed data; and	*See, e.g.,* Crouch, 1989, at pp. 226, 233
[1f] jumping to the position outside the object in the	*See, e.g.,* Crouch, 1989, at pp. 226, 233

Claim Text from '571 Patent	Crouch, 1989
database.	
3. The method of claim 1, wherein the active links are embedded text and wherein the step of selecting comprises activating the embedded text.	*See, e.g.,* Crouch, 1989, at pp. 226, 233
4. The method of claim 1, wherein computer software is used, further comprising: ǀ generating an active link, wherein the active link can be used to jump from a location in the database to another database.	*See, e.g.,* Crouch, 1989, at pp. 226, 233
5. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Crouch, 1989, at p. 234
[5a] choosing a node	*See, e.g.,* Crouch, 1989, at p. 228
[5b] accessing the hyperjump data; ǀ Identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[5c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises proximity analyzing the identified hyperjump data; and	*See, e.g.,* Crouch, 1989, at pp. 228-230
[5d] displaying one or more determined hyperjump data.	*See, e.g.,* Crouch, 1989, at p. 234
6. The method of claim 5, wherein the hyperjump data includes pointers and wherein the direct	*See, e.g.,* Crouch, 1989, at pp. 228, 230, 234

Claim Text from ’571 Patent	Crouch, 1989
reference is a pointer pointing to the chosen node or from the chosen node, and the step of determining comprises analyzing the pointers.	
7. The method of claim 5, wherein the node represents a topic, the determined hyperjump data has a relationship to the topic, and the step of displaying displays determined hyperjump data that has a relationship to the topic.	*See,* above and further disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
8. The method of claim 5, wherein the node is a web page in the network, the accessed hyperjump data are Universal Resource Locators of linked pages, and the step of determining hyperjump data comprises analyzing the identified hyperjump data.	*See, e.g.,* Crouch, 1989, at p. 226
9. The method of claim 5, wherein the node is a document in the network and the determined hyperjump data has a relationship to the document, the step of displaying comprising the step of listing the hyperjump data that has a relationship to the document.	*See, e.g.,* Crouch, 1989, at p. 230
10. The method of claim 5, wherein the step of displaying comprises generating a graphical user display, and wherein information is displayed on a graphical display visually representing more than one coordinate plane.	Disclosed either expressly or inherently in the teachings of the reference and its incorporated disclosures taken as a whole, or in combination with the state of the art at the time of the alleged invention.
11. The method of claim 5, wherein the nodes are nodes in the network that may be accessed, the hyperjump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See, e.g.,* Crouch, 1989, at pp. 226, 233
[11a] generating a source map using one or more of	*See, e.g.,* Crouch, 1989, at pp.226, 230

Claim Text from '571 Patent	Crouch, 1989
the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link; and	
[11b] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	*See, e.g.,* Crouch, 1989, at pp. 226, 233
12. A method for visually displaying data related to a web having identifiable web pages and Universal Resource Locators with pointers, comprising:	*See, e.g.,* Crouch, 1989, at pp.226, 230
[12a] choosing an identifiable web page;	*See, e.g.,* Crouch, 1989, at p. 228
[12b] identifying Universal Resource Locators for the web pages, wherein the identified Universal Resource Locators either point to or point away from the chosen web page;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[12c] analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen web page are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Crouch, 1989, at pp. 228-230
[12d] displaying identities of web pages, wherein the located Universal Resource Locators are used to identify web pages.	*See, e.g.,* Crouch, 1989, at p. 234
13. The method of claim 12, further comprising selecting a web page using the displayed identities of web pages.	*See, e.g.,* Crouch, 1989, at pp. 226, 233

Claim Text from '571 Patent	Crouch, 1989
14. The method of claim 12, further comprising hyperjumping to the selected web page.	*See, e.g.,* Crouch, 1989, at pp. 226, 233
15. The method of claim 12, wherein the step of displaying the identities of web pages comprises generating a graphical user display wherein information within the Universal Resource Locators is parsed and used to generate the graphical user display.	*See, e.g.,* Crouch, 1989, at pp.226, 230
16. A method for navigating documents on the World Wide Web, comprising: l choosing a document;	*See, e.g.,* Crouch, 1989, at p. 228
[16a] identifying documents that have a direct relationship to the chosen document;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[16b] locating documents that have an indirect relationship to the chosen document identifying Universal Resource Locators for the documents, wherein the identified Universal Resource Locators either point to or point away from the chosen document; analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen document are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Crouch, 1989, at pp. 228, 229
[16c] displaying a located document.	*See, e.g.,* Crouch, 1989, at p. 234, 228, 229

Claim Text from ’571 Patent	Crouch, 1989
17. The method of claim 16, wherein pages and their respective Universal Resource Locators are used and the step of locating documents comprises analyzing the pages and their respective Universal Resource Locators.	*See, e.g.,* Crouch, 1989, at p. 226
18. The method of claim 17, wherein the step of analyzing pages comprises cluster analyzing the pages.	*See, e.g.,* Crouch, 1989, at pp. 228, 229
19. The method of claim 16, wherein the step of displaying a located document comprises:	*See, e.g.,* Crouch, 1989, at p. 233, 234
[19a] generating a screen display of identities of one or more located documents; and	*See, e.g.,* Crouch, 1989, at p. 234
[19b] selecting one or more of the located documents.	*See, e.g.,* Crouch, 1989, at pp. 226, 233
20. The method of claim 19, wherein the step of generating a screen display comprises generating a graphical display.	*See, e.g.,* Crouch, 1989, at pp.226, 230
21. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Crouch, 1989, at pp.233, 234
[21a] choosing a node;	*See, e.g.,* Crouch, 1989, at p. 228
[21b] accessing the hyperjump data; | identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[21c] determining hyperjump data from within the	*See, e.g.,* Crouch, 1989, at pp. 228, 229

Claim Text from '571 Patent	Crouch, 1989
accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises cluster analyzing the hyperjump data; and	
[21d] displaying one or more determined hyperjump data.	*See, e.g.,* Crouch, 1989, at p. 233, 234
22. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Crouch, 1989, at p. 234
[22a] choosing a node;	*See, e.g.,* Crouch, 1989, at p. 228
[22b] accessing the hyperjump data; | identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Crouch, 1989, at pp. 228, 230
[22c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data; and	*See, e.g.,* Crouch, 1989, at pp. 228-230
[22d] displaying one or more determined hyperjump data, wherein the nodes are nodes in the network that may be accessed, the hypejump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See, e.g.,* Crouch, 1989, at pp. 233, 234
[22e] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link, and wherein the method further comprises activating a link represented on the source map, wherein a user may	*See, e.g.,* Crouch, 1989, at pp. 228, 230, 233

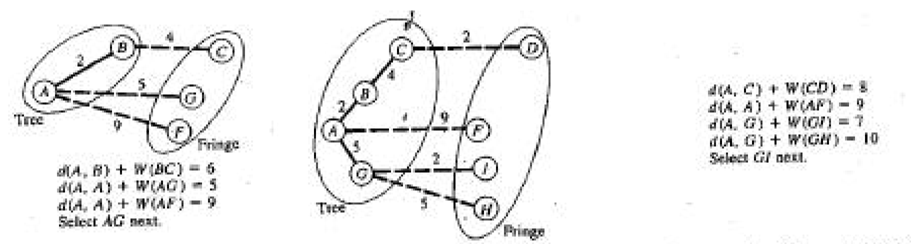
Claim Text from ’571 Patent	Crouch, 1989
hyperjump to a node represented as a node of the link.	

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 6,233,571

Based on Baase, S., Computer Algorithms: Introduction to Design and Analysis, 2nd Edition, Addison-Wesley Publishing Co., 1988. (“Baase, 1988)

Claim Text from ’571 Patent	Baase, 1988
5. A method for displaying information about a network that has hyperjump data, comprising:	*See* p. 149-156 and 167-72, Title (Computer Algorithms).
[5a] Choosing a node	*See, e.g.,* Baase, 1988, at p. 149-156, 160-67 and 168-172, Djisktra’s shortest path algorithm will find shortest paths from v to the other vertices in order of increasing distance from v. . . . The algorithm starts at one vertex (v) and “branches out” by selecting certain edges that lead to new vertices (p. 168) x:= v (p. 171).
[5b] accessing the hyperjump data; ǀ Identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Baase, 1988, at p. 160-67, 168-172, 184-91, 193-97.  (b) An intermediate step. (c) An intermediate step. (*CH* was considered but not chosen to replace *GH* as a candidate.) Whether or not G is a digraph, it is helpful 10 think of the tree and candidate edges as having an orientation; the tail of an edge is the vertex closer to *v*. Candidate edges go from a tree vertex to a fringe vertex. These edges will always be written to reflect this orientation; in other words, if we write *XY*. we are assuming that *x* is closer to *v* than y is. We will refer *tox* as *tail(xy)* and y as *head(xy)* even if G is not a directed graph.
[5c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises proximity analyzing the identified hyperjump data; and	

Claim Text from '571 Patent	Baase, 1988
	Given the situation in Fig. 4.18(c), the next step is to select a candidate edge and fringe vertex. We choose a candidate edge *e* for which *d(v,tail(e))+W(e)* is minimum. This is the weight of the path obtained by adjoining *e* to the known shortest path to *tail(e).* Since the quantity *d(v, tail(e))+W(e)* for a candidate edge *e* may be used repeatedly, it can be computed once and saved. To compute it efficiently when *e*first becomes a candidate, we also save *d(v,y)* for each y in the tree. Thus we use an array *dist* as follows: dist[y] = d(v,y); dist[z] = d(v,y) + W(yz). After a vertex and the corresponding candidate edge are selected, the information in the data structure must be updated. In Fig. 4.18(d) the vertex *I* and the edge *Gl* have just been selected. The candidate edge for *F* was *AF,* but now *AF* must be replaced by *IF* because *IF* yields a shorter path to *F,* We must also recompute *dist[F]*. The vertex *E,* which was unseen, is now on the fringe because it is adjacent to I, now in the tree . . . while x ≠ w and not *stuck* do . . . . end { while x ≠ w and not *stuck* } (p. 171-172)
[5d] displaying one or more determined hyperjump data.	*See, e.g.,* Baase, 1988, at p. 167, 168-172, including e.g. {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)
6. The method of claim 5, wherein the hyperjump data includes pointers and wherein the direct reference is a pointer pointing to the chosen node or from the chosen node, and the step of determining comprises analyzing the pointers.	*See, e.g.,* Baase, 1988, at p. 149, 167, 168-172.
7. The method of claim 5, wherein the node represents a topic, the determined hyperjump data has a relationship to the topic, and the step of displaying displays determined hyperjump data that	*See, e.g.,* Baase, 1988, at p. 149, 167, 168-172, 175-76.

Claim Text from '571 Patent	Baase, 1988
has a relationship to the topic.	
12. A method for visually displaying data related to a web having identifiable web pages and Universal Resource Locators with pointers, comprising:	*See* p. 149-156 and 167-72,
[12a] choosing an identifiable web page;	*See, e.g.,* Baase, 1988, at p. 149-156, 160-161 and 168-172, Djisktra's shortest path algorithm will find shortest paths from v to the other vertices in order of increasing distance from v. . . . The algorithm starts at one vertex (v) and "branches out" by selecting certain edges that lead to new vertices (p. 168) x:= v (p. 171).
[12b] identifying Universal Resource Locators for the web pages, wherein the identified Universal Resource Locators either point to or point away from the chosen web page;	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.
[12c] analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen web page are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.
[12d] displaying identities of web pages, wherein the located Universal Resource Locators are used to identify web pages.	*See, e.g.,* Baase, 1988, at p. 167, 168-172, including e.g. {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)b

Claim Text from '571 Patent	Baase, 1988
16. A method for navigating documents on the World Wide Web, comprising: ǀ choosing a document;	
[16a] Identifying documents that have a direct relationship to the chosen document;	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.
[16b] locating documents that have an indirect relationship to the chosen document identifying Universal Resource Locators for the documents, wherein the identified Universal Resource Locators either point to or point away from the chosen document; analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen document are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.
[16c] displaying a located document.	*See, e.g.,* Baase, 1988, at p. 167, 168-172, including e.g. {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)
18. The method of claim 17, wherein the step of analyzing pages comprises cluster analyzing the pages.	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97
21. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Baase, 1988, p. 149-156 and 167-72, Title

Claim Text from ’571 Patent	Baase, 1988
[21a] choosing a node;	*See, e.g.,* Baase, 1988, at p. 149-156 and 168-172
[21b] accessing the hyperjump data;	identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.
[21c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises cluster analyzing the hyperjump data; and	*See, e.g.,* Baase, 1988, at p. 160-167, 168-172, 175-76, 184-91, 193-97.
[21d] displaying one or more determined hyperjump data.	*See, e.g.,* Baase, 1988, at p. 168-172. {Output the path, the vertices will be listed in the reverse order, i.e. from w to v} While x ≠ 0 do Output(x); x:= parent[x] end (p. 172)

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 6,233,571

BASED ON BOTAFOGO, R.A. ET AL. "IDENTIFYING AGGREGATES IN HYPERTEXT STRUCTURES" HYPERTEXT '91 PROCEEDINGS, 63-74 (1991). ("BOTAFOGO, 1991")

Claim Text from '571 Patent	Botafogo, 1991
1. A method for using active links within the data of an object stored in a database of a computer so that a user may jump from viewing the data of the object in the database to a position outside the object in the database and outside the computer, comprising:	*See, e.g.,* Botafogo, 1991, at p. 63-66, 70-72
[1a] storing one or more links within data of the object in the database to positions outside of the computer, wherein the stored links are active links;	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72
[1b] displaying the data of the object within the database, wherein one or more active links are displayed with the data from the object in the database, wherein positions are nodes in a network that may be accessed, the active links including hyperjump links between nodes in the network and the objects, and the step of displaying comprises:	*See, e.g.,* Botafogo, 1991, at p. 71-72
[1c] generating a source map, wherein the source map represents hyperjump links that identify a chosen node as a destination of a link, and	*See, e.g.,* Botafogo, 1991, at p. 71-72, Fig. 4,5.
[1d] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link;	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72
[1e] selecting one of the displayed active links from those displayed with the displayed data; and	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72

Claim Text from '571 Patent	Botafogo, 1991
[1f] jumping to the position outside the object in the database.	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72
3. The method of claim 1, wherein the active links are embedded text and wherein the step of selecting comprises activating the embedded text.	*See, e.g.,* Botafogo, 1991, at 63-64
4. The method of claim 1, wherein computer software is used, further comprising: | generating an active link, wherein the active link can be used to jump from a location in the database to another database.	*See, e.g.,* Botafogo, 1991, at 63-64
5. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Botafogo, 1991, at p. 63-64, 71-72
[5a] choosing a node	*See, e.g.,* Botafogo, 1991, at p. 64-65
[5b] accessing the hyperjump data; | Identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Botafogo, 1991, at 64-65, 66, 68, 70
[5c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises proximity analyzing the identified hyperjump data; and	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
[5d] displaying one or more determined hyperjump data.	*See, e.g.,* Botafogo, 1991, at p. 71-72

Claim Text from ’571 Patent	Botafogo, 1991
6. The method of claim 5, wherein the hyperjump data includes pointers and wherein the direct reference is a pointer pointing to the chosen node or from the chosen node, and the step of determining comprises analyzing the pointers.	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
7. The method of claim 5, wherein the node represents a topic, the determined hyperjump data has a relationship to the topic, and the step of displaying displays determined hyperjump data that has a relationship to the topic.	*See, e.g.,* Botafogo, 1991, at p. 66-67, 71-72
8. The method of claim 5, wherein the node is a web page in the network, the accessed hyperjump data are Universal Resource Locators of linked pages, and the step of determining hyperjump data comprises analyzing the identified hyperjump data.	*See, e.g.,* Botafogo, 1991, at p. 66-67, 71-72
9. The method of claim 5, wherein the node is a document in the network and the determined hyperjump data has a relationship to the document, the step of displaying comprising the step of listing the hyperjump data that has a relationship to the document.	*See, e.g.,* Botafogo, 1991, at p. 71-72
10. The method of claim 5, wherein the step of displaying comprises generating a graphical user display, and wherein information is displayed on a graphical display visually representing more than one coordinate plane.	*See, e.g.,* Botafogo, 1991, at p. 71-72
11. The method of claim 5, wherein the nodes are nodes in the network that may be accessed, the hyperjump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See below*

Claim Text from '571 Patent	Botafogo, 1991
[11a] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link; and	*See, e.g.,* Botafogo, 1991, at p. 71-72
[11b] wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	*See, e.g.,* Botafogo, 1991, at p. 71-72
12. A method for visually displaying data related to a web having identifiable web pages and Universal Resource Locators with pointers, comprising:	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72
[12a] choosing an identifiable web page;	*See, e.g.,* Botafogo, 1991, at p. 64-65
[12b] identifying Universal Resource Locators for the web pages, wherein the identified Universal Resource Locators either point to or point away from the chosen web page;	*See, e.g.,* Botafogo, 1991, at 64-65, 66, 68, 70
[12c] analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen web page are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
[12d] displaying identities of web pages, wherein the located Universal Resource Locators are used to identify web pages.	*See, e.g.,* Botafogo, 1991, at p. 71-72
13. The method of claim 12, further comprising selecting a web page using the displayed identities	*See, e.g.,* Botafogo, 1991, at p. 71-72

Claim Text from ’571 Patent	Botafogo, 1991
of web pages.	
14. The method of claim 12, further comprising hyperjumping to the selected web page.	*See, e.g.,* Botafogo, 1991, at p. 63-64, 71-72
15. The method of claim 12, wherein the step of displaying the identities of web pages comprises generating a graphical user display wherein information within the Universal Resource Locators is parsed and used to generate the graphical user display.	*See, e.g.,* Botafogo, 1991, at p. 71-72
16. A method for navigating documents on the World Wide Web, comprising: | choosing a document;	*See, e.g.,* Botafogo, 1991, at p. 63-64, 71-72
[16a] identifying documents that have a direct relationship to the chosen document;	*See, e.g.,* Botafogo, 1991, at 64-65, 66, 68, 70
[16b] locating documents that have an indirect relationship to the chosen document identifying Universal Resource Locators for the documents, wherein the identified Universal Resource Locators either point to or point away from the chosen document; analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen document are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
[16c] displaying a located document.	*See, e.g.,* Botafogo, 1991, at p. 71-72

Claim Text from ’571 Patent	Botafogo, 1991
17. The method of claim 16, wherein pages and their respective Universal Resource Locators are used and the step of locating documents comprises analyzing the pages and their respective Universal Resource Locators.	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
18. The method of claim 17, wherein the step of analyzing pages comprises cluster analyzing the pages.	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
19. The method of claim 16, wherein the step of displaying a located document comprises:	*See, e.g.,* Botafogo, 1991, at p. 71-72
[19a] generating a screen display of identities of one or more located documents; and	*See, e.g.,* Botafogo, 1991, at p. 71-72
[19b] selecting one or more of the located documents.	*See, e.g.,* Botafogo, 1991, at p. 71-72
20. The method of claim 19, wherein the step of generating a screen display comprises generating a graphical display.	*See, e.g.,* Botafogo, 1991, at p. 71-72
21. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72
[21a] choosing a node;	*See, e.g.,* Botafogo, 1991, at p. 64-65
[21b] accessing the hyperjump data; | identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the	*See, e.g.,* Botafogo, 1991, at 64-65, 66, 68, 70

Claim Text from '571 Patent	Botafogo, 1991
chosen node;	
[21c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises cluster analyzing the hyperjump data; and	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
[21d] displaying one or more determined hyperjump data.	*See, e.g.,* Botafogo, 1991, at p. 71-72
22. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.,* Botafogo, 1991, at p. 63, 71-72
[22a] choosing a node;	*See, e.g.,* Botafogo, 1991, at p. 64-65
[22b] accessing the hyperjump data; ǀ identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* Botafogo, 1991, at 64-65, 66, 68, 70
[22c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data; and	*See, e.g.,* Botafogo, 1991, at p. 64-65, 66, 68, 70
[22d] displaying one or more determined hyperjump data, wherein the nodes are nodes in the network that may be accessed, the hypejump data includes hyperjump links between nodes in the network, and the step of displaying comprises:	*See, e.g.,* Botafogo, 1991, at p. 71-72

Claim Text from ’571 Patent	Botafogo, 1991
[22e] generating a source map using one or more of the determined hyperjump data, wherein the source map represents hyperjump links that identify the chosen node as a destination of a link, and wherein the method further comprises activating a link represented on the source map, wherein a user may hyperjump to a node represented as a node of the link.	*See, e.g.,* Botafogo, 1991, at p. 71-72

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 6,233,571

- **Based on Joachims, T et al., "WebWatcher: Machine Learning and Hypertext" Proceedings of the 1995 AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, 1995 ("Joachims 1995 1995")**

Claim Text from '571 Patent	Joachims 1995
5. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.*, p. 1-4.
[5a] Choosing a node	*See, e.g.*, p.2, 4 (e.g. "The target function we want to learn is a mapping from an arbitrary web page to a set of related pages.").
[5b] accessing the hyperjump data; | Identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.*, p. 3-4 (e.g., "3.3 Algorithm")
[5c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises proximity analyzing the identified hyperjump data; and	*See, e.g.*, p. 4-5 (e.g., "3.3 Algorithm")
[5d] displaying one or more determined hyperjump data.	*See, e.g.*, p. 2, 4 (e.g., "The pages associated with the n most similar columns are returned by Related.").
6. The method of claim 5, wherein the hyperjump data includes pointers and wherein the direct reference is a pointer pointing to the chosen node or from the chosen node, and the step of determining comprises analyzing the pointers.	*See, e.g.*, p. 4 (e.g., "3.3 Algorithm")
7. The method of claim 5, wherein the node represents a topic, the determined hyperjump data has a relationship to the topic, and the step of displaying displays determined hyperjump data that has a relationship to the topic.	*See, e.g.*, p. 4 (e.g., "3.3 Algorithm")
8. The method of claim 5, wherein the node is a web	*See e.g.*, p. 1-3.

Claim Text from ’571 Patent	Joachims 1995
page in the network, the accessed hyperjump data are Universal Resource Locators of linked pages, and the step of determining hyperjump data comprises analyzing the identified hyperjump data.	
9. The method of claim 5, wherein the node is a document in the network and the determined hyperjump data has a relationship to the document, the step of displaying comprising the step of listing the hyperjump data that has a relationship to the document.	*See, e.g.,* p. 4
12. A method for visually displaying data related to a web having identifiable web pages and Universal Resource Locators with pointers, comprising:	*See, e.g.,* p. 1, 3-5
[12a] choosing an identifiable web page;	*See, e.g.,* p.2, 4 (e.g. “The target function we want to learn is a mapping from an arbitrary web page to a set of related pages.”).
[12b] identifying Universal Resource Locators for the web pages, wherein the identified Universal Resource Locators either point to or point away from the chosen web page;	*See, e.g.,* p. 3-4 (e.g., “3.3 Algorithm”)
[12c] analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen web page are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.,* p. 4-5 (e.g., “3.3 Algorithm”)
[12d] displaying identities of web pages, wherein the located Universal Resource Locators are used to identify web pages.	*See, e.g.,* p. 2, 4 (e.g., “The pages associated with the n most similar columns are returned by Related.”).
13. The method of claim 12, further comprising selecting a web page using the displayed identities of web pages.	*See, e.g.,* p. 2, 4 (e.g., “The pages associated with the n most similar columns are returned by Related.”).
14. The method of claim 12, further comprising	*See, e.g.,* p. 1-3

Claim Text from ’571 Patent	Joachims 1995
hyperjumping to the selected web page.	
15. The method of claim 12, wherein the step of displaying the identities of web pages comprises generating a graphical user display wherein information within the Universal Resource Locators is parsed and used to generate the graphical user display.	*See, e.g.*, p. 4
16. A method for navigating documents on the World Wide Web, comprising: ǀ choosing a document;	*See, e.g.*, p. 1, 3-5
[16a] Identifying documents that have a direct relationship to the chosen document;	*See, e.g.*, p. 4
[16b] locating documents that have an indirect relationship to the chosen document identifying Universal Resource Locators for the documents, wherein the identified Universal Resource Locators either point to or point away from the chosen document; analyzing Universal Resource Locators, including the identified Universal Resource Locators, wherein Universal Resource Locators which have an indirect relationship to the chosen document are located, wherein the step of analyzing further comprises cluster analyzing the Universal Resource Locators for indirect relationships; and	*See, e.g.*, p. 4
[16c] displaying a located document.	*See, e.g.*, p. 4 (e.g., “The pages associated with the n most similar columns are returned by Related.”).
17. The method of claim 16, wherein pages and their respective Universal Resource Locators are used and the step of locating documents comprises analyzing the pages and their respective Universal Resource Locators.	*See, e.g.*, p. 1, 4-5
18. The method of claim 17, wherein the step of analyzing pages comprises cluster analyzing the pages.	*See, e.g.*, p. 1-4.
19. The method of claim 16, wherein the step of	

Claim Text from ’571 Patent	Joachims 1995
displaying a located document comprises:	
[19a] generating a screen display of identities of one or more located documents; and	*See, e.g.*, p. 2-3
[19b] Selecting one or more of the located documents.	*See, e.g.*, p. 2-3
20. The method of claim 19, wherein the step of generating a screen display comprises generating a graphical display.	*See, e.g.*, p. 1-4.
21. A method for displaying information about a network that has hyperjump data, comprising:	*See, e.g.*, p. 1-4.
[21a] choosing a node;	*See, e.g.,* p. 2, 4 (e.g. “The target function we want to learn is a mapping from an arbitrary web page to a set of related pages.”).
[21b] accessing the hyperjump data; ǀ identifying hyperjump data from within the accessed hyperjump data that has a direct reference to the chosen node;	*See, e.g.,* p. 3-4 (e.g., “3.3 Algorithm”)
[21c] determining hyperjump data from within the accessed hyperjump data that has an indirect reference to the chosen node using the identified hyperjump data, wherein the step of determining comprises cluster analyzing the hyperjump data; and	*See, e.g.,* p. 4-5 (e.g., “3.3 Algorithm”)
[21d] displaying one or more determined hyperjump data.	*See, e.g.,* p. 2, 4 (e.g., “The pages associated with the n most similar columns are returned by Related.”).

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court’s construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff’s Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants’ accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.

INVALIDITY CLAIM CHART FOR U.S. PATENT NO. 6,233,571

Based on Caplinger, M., "Graphical Database Browsing," ACM p. 113-121 (1986)

Claim Text from '571 Patent	Calpinger 1986
10. The method of claim 5, wherein the step of displaying comprises generating a graphical user display, and wherein information is displayed on a graphical display visually representing more than one coordinate plane.	*See, e.g.,* p. 118 (e.g., "6.1 3D Browser")

Defendants reserve the right to revise this contention chart concerning the invalidity of the asserted claims, as appropriate, for example depending upon the Court's construction of the asserted claims, any findings as to the priority date of the asserted claims, and/or positions that Plaintiff or its expert witness(es) may take concerning claim interpretation, construction, infringement, and/or invalidity issues.

Plaintiff's Infringement Contentions are based on an apparent construction of the claim terms. Defendants disagree with these apparent constructions. Nothing stated herein shall be treated as an admission or suggestion that Defendants agree with Plaintiff regarding either the scope of any of the asserted claims or the claim constructions advanced by Plaintiff in its Infringement Contentions or anywhere else, or that any of Defendants' accused technology meets any limitations of the claims. Nothing stated herein shall be construed as an admission or a waiver of any particular construction of any claim term. Defendants also reserve all their rights to challenge any of the claim terms herein under 35 U.S.C. § 112, including by arguing that they are indefinite, not supported by the written description and/or not enabled. Accordingly, nothing stated herein shall be construed as a waiver of any argument available under 35 U.S.C. § 112.